

# **PCI Express Architecture Platform Init/Config**

## **Test Specification**

Revision 3.0

June 18, 2013

---



Revision	Issue Date	Comments
1.0	05/16/2004	Initial Release.
3.0	06/06/2013	Updated to reflect PCI Express Base Specification, Revision 3.0

---

PCI-SIG® disclaims all warranties and liability for the use of this document and the information contained herein and assumes no responsibility for any errors that may appear in this document, nor does PCI-SIG make a commitment to update the information contained herein.

Contact the PCI-SIG office to obtain the latest revision of this specification.

Questions regarding this specification or membership in PCI-SIG may be forwarded to:

#### **Membership Services**

[www.pcisig.com](http://www.pcisig.com)

E-mail: [administration@pcisig.com](mailto:administration@pcisig.com)

Phone: 503-619-0569

Fax: 503-644-6708

#### **Technical Support**

[techsupp@pcisig.com](mailto:techsupp@pcisig.com)

#### **DISCLAIMER**

This PCI Code and ID Assignment Specification is provided *as is* with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. PCI-SIG disclaims all liability for infringement of proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

PCI Express is a trademark of PCI-SIG.

All other product names are trademarks, registered trademarks, or servicemarks of their respective owners.

© 2012, 2013 PCI SIG. All rights reserved.

<b>CONTENTS</b>		
	<b>2.1. Test Topologies and Hardware Functionality .....</b>	<b>30</b>
1.	<b>INTRODUCTION .....</b>	<b>29</b>
2.1.1.	Platform Init/Config Tests .....	30
2.	<b>TEST DESCRIPTIONS .....</b>	<b>30</b>
2.1.2.	General Topology Test Checks .....	30
2.1.2.1.	Memory BAR Configuration .....	31
2.1.2.2.	IO BAR Configuration .....	32
2.1.2.3.	Expansion ROM Configuration .....	32
2.1.2.4.	Bridge Address Assignment .....	34
2.1.2.5.	Bridge Memory and IO Window Configuration .....	35
	<b>2.2. Devices in Test Topologies .....</b>	<b>36</b>
2.2.1.	Default registers and capability structures .....	36
2.2.2.	Default PCI Express Devices .....	36
2.2.2.1.	Default PCI Express Endpoint .....	36
2.2.2.2.	Default PCI Express Legacy Endpoint .....	37
2.2.2.3.	Default PCI Express Upstream Switch Port .....	38
2.2.2.4.	Default PCI Express Downstream Switch Port .....	38
2.2.2.5.	Default PCI Express Downstream Switch Port with Slot .....	39
2.2.2.6.	Default PCI Express Downstream Switch Port with Hot Plug Slot .....	39
2.2.3.	Modifications to Default Devices .....	40
2.2.3.1.	Device Type .....	40
2.2.3.2.	Base Address Registers .....	40
2.2.3.3.	Default Register Values .....	41
2.2.3.4.	Register Characteristics .....	43
2.2.3.5.	Capabilities (Adding or Removing) .....	43
2.2.3.6.	Bridges and Downstream Devices .....	44
2.2.3.7.	Switches .....	44
2.2.3.8.	Expansion Rom .....	46
2.2.3.9.	Command Register Enable Bits .....	46
2.2.3.10.	Identical Devices .....	47
2.2.3.11.	Device Response Characteristics .....	47
2.2.3.12.	Active State Power Management .....	48
2.2.3.13.	Vendor ID and Device ID .....	49

<b>3.1. Tests 1.x - Worst Case Complex Topology Support Test Cases .....</b>	<b>50</b>
3.1.1. Test 1.1 - Null Case - Informational Only Test.....	50
3.1.1.1. Starting Configuration.....	50
3.1.1.2. Overview of Test Steps .....	60
3.1.1.3. Results Interpretation .....	60
3.1.2. Test 1.2 – Variety of Endpoint Requests - Informational Only Test .....	61
3.1.2.1. Starting Configuration.....	61
<b>3.2. Tests 2.x - PCI-SIG PCI Functional Bios Test Cases .....</b>	<b>76</b>
3.2.1. Test 2.1 - Null Case – Required Test .....	76
3.2.1.1. Starting Configuration.....	76
3.2.1.2. Overview of Test Steps .....	77
3.2.1.3. Results Interpretation .....	77
3.2.2. Test 2.2, Simple Case – Required Test.....	78
3.2.3. Base 1.1 Topology .....	78
3.2.3.1. Starting Configuration.....	78
3.2.4. Base 2.0 Topology .....	78
3.2.4.1. Starting Configuration.....	78
3.2.5. Base 3.0 Topology .....	79
3.2.5.1. Starting Configuration.....	79
3.2.6. Test 2.3: Simple Case with Gaps – Required Test.....	80
3.2.7. Test 2.4 – A lot of I/O stuff – Required Test .....	80
3.2.8. Test 2.5: More I/O Stuff (Checks a lot of 256B I/O requests – Required Test .....	81
3.2.9. Test 2.6: A Bunch Asking for 256B Memory – Required Test .....	82
3.2.10. Test 2.7 – Memory Requests of Various Sizes – Required Test .....	82
3.2.11. Test 2.8 – Memory Requests of Various Sizes with Prefetchable Bit Set – Required Test .....	83
3.2.12. Test 2.9: I/O Requests with top 16 Bits Tied to Zero – Required Test.....	84
3.2.13. Test 2.10: 64-Bit BARs – Required Test.....	85
3.2.14. Test 2.11: 64-Bit BARs with Prefetchable Bit Set – Required Test.....	85
3.2.15. Test 2.12: Four BARs Asking for 1GB – Required Test.....	86
3.2.16. Test 2.13: I/O Registers Asking for More Than 256 Bytes – Required Test .....	87
3.2.17. Test 2.14: One Function Requests Memory of Various Sizes While the Other Makes No Requests – Required Test .....	88
3.2.18. Test 2.15: One Function Requests Several GB of Memory While the Other Makes Reasonable Memory Requests – Required Test .....	89
3.2.19. Test 2.16: Simple Expansion ROM Case – Optional Test .....	90
3.2.20. Test 2.17: Function 0 Should Run; Function 1 Should not – Optional Test .....	90

3.2.21.	Test 2.18: Function 1 Should Run; Function 0 Should Not – Optional Test.....	91
3.2.22.	Test 2.19: Both Functions Run – Optional Test.....	92
3.2.23.	Test 2.20: Two 34KB Images – Optional Test .....	93
3.2.24.	Test 2.21: Two 34KB Images, ONE SHRINKS to 5KB RUNTIME–Optional Test.....	94
3.2.25.	Test 2.22 – Two 34KB Images, Other Shrinks to 5KB Runtime – Optional Test .....	95
3.2.26.	Test 2.23: First Function with Zero Runtime Size – Optional Test.....	96
3.2.27.	Test 2.24: Second Function with Zero Runtime Size – Optional Test .....	97
3.2.28.	Test 2.25: Two Different Images – Optional Test .....	98
<b>3.3.</b>	<b>Tests 3.x - Expanded Functional Test Cases.....</b>	<b>99</b>
3.3.1.	Test 3.1 - Nine Port Switch with No Requests –OptionalTest.....	99
3.3.1.1.	Starting Configuration.....	99
3.3.1.2.	Overview of Test Steps .....	100
3.3.1.3.	Results Interpretation .....	100
3.3.2.	Test 3.2: Nine Port Switch with Various Requests – Optional Test .....	101
3.3.2.1.	Starting Configuration.....	101
3.3.3.	Test 3.3: Bridge with IO Requests - Optional Test .....	105
3.3.3.1.	Starting Configuration.....	105
3.3.4.	Test 3.4: Bridge with Memory Requests - Optional Test .....	106
3.3.4.1.	Starting Configuration.....	106
3.3.5.	Test 3.5: Bridge with 64 Bit Memory Request - Optional Test .....	106
3.3.5.1.	Starting Configuration.....	106
3.3.6.	Test 3.6: Bridge with Expansion ROM - Optional Test .....	107
3.3.6.1.	Starting Configuration.....	107
3.3.7.	Test 3.7 – Multiple Bridges with Various Requests - Informational Only Test ...	107
3.3.7.1.	Starting Configuration.....	107
3.3.8.	Test 3.8: Eight Function Device with Multiple Requests - Informational Only Test.....	116
3.3.8.1.	Starting Configuration.....	116
3.3.9.	Test 3.9: Eight Function Device with Type 0 and Type 1 Headers No Resource Requests - Informational Only Test .....	119
3.3.9.1.	Starting Configuration.....	119
3.3.10.	Test 3.10: Eight Function Device with Type 0 and Type 1 Headers Various Resource Requests - Informational Only Test .....	122
3.3.10.1.	Starting Configuration.....	122
3.3.11.	Test 3.11: Five Levels of Bridges without Resource Requests - Informational Only Test.....	127
3.3.11.1.	Starting Configuration.....	127
3.3.12.	Test 3.12: Eight Function Device with Only Last Function Implemented - Informational Only Test.....	131
3.3.12.1.	Starting Configuration.....	131

3.3.13. Test 3.13: Eight Function Device with Alternating Functions Implemented - Informational Only Test.....	132
3.3.13.1. Starting Configuration.....	132
<b>3.4. Test 4: Legacy Interrupt Support - Informational Only Test .....</b>	<b>133</b>
3.4.1. Starting Configuration.....	133
3.4.2. Overview of Test Steps .....	133
3.4.3. Results Interpretation .....	134
<b>3.5. Test 5: Device Ignores PME Fence Mechanism - Informational Only Test.....</b>	<b>134</b>
3.5.1. Starting Configuration.....	134
3.5.2. Overview of Test Steps .....	134
3.5.3. Results Interpretation .....	135
<b>3.6. Tests 6.x: Extended Expansion ROM Test Cases .....</b>	<b>135</b>
3.6.1. Test 6.1: Device List Pointer - Informational Only Test.....	135
3.6.1.1. Starting Configuration.....	135
3.6.1.2. Overview of Test Steps .....	136
3.6.1.3. Results Interpretation .....	136
3.6.2. Test 6.2: Maximum Run Time Length - Informational Only Test .....	136
3.6.2.1. Starting Configuration.....	136
3.6.2.2. Overview of Test Steps .....	137
3.6.2.3. Results Interpretation .....	137
<b>3.7. Tests 7.x - Hot Plug Support Test Cases.....</b>	<b>138</b>
3.7.1. Test 7.1: Max Device Null Case - Informational Only Test .....	138
3.7.1.1. Starting Configuration.....	138
3.7.1.2. Overview of Test Steps .....	141
3.7.1.3. Results Interpretation .....	141
3.7.2. Test 7.2: Max Devices – Various Requests - Informational Only Test.....	142
3.7.2.1. Starting Configuration.....	142
3.7.3. Test 7.3: Nine Port Switch Null Case - Informational Only Test .....	150
3.7.3.1. Starting Configuration.....	150
3.7.4. Test 7.4: Nine Port Switch with Multiple Requests- Informational Only Test .....	152
3.7.4.1. Starting Configuration.....	152
3.7.5. Test 7.5: Simple Five Level Switch Case - Informational Only Test .....	156
3.7.5.1. Starting Configuration.....	156
3.7.6. Test 7.6: Five Levels of Bridges with Various Requests - Informational Only Test.....	158
3.7.6.1. Starting Configuration.....	158

<b>3.8. Tests 8.x: L0s Active State PM Enabling In Complex Topologies Test Cases....</b>	<b>166</b>
3.8.1. Test 8.1: Single Device with Latency Mismatch - Informational Only Test .....	166
3.8.1.1. Starting Configuration.....	166
3.8.1.2. Overview of Test Steps .....	166
3.8.1.3. Results Interpretation .....	167
3.8.2. Test 8.2: Multi-Function Device with Latency Mismatch in One Function - Informational Only Test.....	167
3.8.2.1. Starting Configuration.....	167
3.8.3. Test 8.3: One Device and One Switch Have Mismatches in Large Topology - Informational Only Test.....	168
3.8.3.1. Starting Configuration.....	168
3.8.4. Test 8.4: One Switch in Path Creates Latency Mismatch - Informational Only Test	171
3.8.4.1. Starting Configuration.....	171
<b>3.9. Tests 9.x - L1 Active State PM Enabling In Complex Topologies Test Cases.....</b>	<b>174</b>
3.9.1. Test 9.1: Single Device with Latency Mismatch - Informational Only Test .....	174
3.9.1.1. Starting Configuration.....	174
3.9.1.2. Overview of Test Steps .....	175
3.9.1.3. Results Interpretation .....	175
3.9.2. Test 9.2: Multi-Function Device with Latency Mismatch in One Function - Informational Only Test.....	175
3.9.2.1. Starting Configuration.....	175
3.9.3. Test 9.3: One Device and One Switch Have Mismatches in Large Topology - Informational Only Test.....	176
3.9.3.1. Starting Configuration.....	176
3.9.4. Test 9.4 - One Switch in Path Creates Latency Mismatch - Informational Only Test.....	179
3.9.4.1. Starting Configuration.....	179
3.9.5. Test 9.5: Two Switch Delays Cause Potential L1 Mismatch - Informational Only Test.....	182
3.9.5.1. Starting Configuration.....	182
3.9.6. Test 9.6: Four Switch Delays Cause Potential L1 Mismatch - Informational Only Test.....	184
3.9.6.1. Starting Configuration.....	184
<b>3.10. Test 10: Worst Case Device Response Time to Configuration Requests After       Reset - Informational Only Test .....</b>	<b>186</b>
3.10.1. Starting Configuration.....	186
3.10.2. Overview of Test Steps .....	186
3.10.3. Results Interpretation .....	187

<b>3.11. Test 11 - Worst Case Device Entry into Link Training After Reset - Informational Only Test.....</b>	<b>187</b>
3.11.1. Starting Configuration.....	187
3.11.2. Overview of Test Steps .....	187
3.11.2.1. Results Interpretation .....	188
<b>3.12. Test 12: Platform Response to Configuration Retry Responses - Informational Only Test.....</b>	<b>188</b>
3.12.1. Starting Configuration.....	188
3.12.2. Overview of Test Steps .....	188
3.12.3. Results Interpretation .....	189
<b>3.13. Test 13: Platform Response to “Broken” Device during Configuration - Informational Only Test .....</b>	<b>190</b>
3.13.1. Starting Configuration.....	190
3.13.2. Overview of Test Steps .....	191
3.13.3. Results Interpretation .....	192
<b>3.14. Tests 14.x: Bios/OS Interaction Standard Test Cases .....</b>	<b>192</b>
3.14.1. Test 14.1: MMCFG Reporting Of Extended Config Access Base Address - Informational Only Test.....	192
3.14.2. Test 14.2: OHSP Method for Hot Plug Capable Systems - Informational Only Test.....	192
<b>3.15. Test 15: Max_Payload_Size Test – Required Test.....</b>	<b>193</b>
<b>3.16. Test 16: End-End TLP Prefix Test – Informational-only Test.....</b>	<b>193</b>
<b>3.17. Test 17: MFVC Capability Test – Required Test .....</b>	<b>194</b>
<b>3.18. Test 18: VC Mapping Test – Required Test.....</b>	<b>194</b>
<b>3.19. Test 19: PME Enable Test – Required Test .....</b>	<b>195</b>
<b>3.20. Test 20 Slot Power Test – Required Test .....</b>	<b>195</b>
<b>3.21. Test 21 LTR Mechanism Test – Required Test .....</b>	<b>196</b>
<b>3.22. Test 22 ASPM Optionality Test – Required Test.....</b>	<b>196</b>
<b>3.23. Test 23: Common Clock Tests .....</b>	<b>197</b>
3.23.1. Test 23a: Common Clock Test 1 – Required Test.....	197
3.23.2. Test 23B: Common Clock Test 2 – Required Test .....	198
<b>3.24. Test 24: Clock Power Management Tests .....</b>	<b>199</b>
3.24.1. Test 24a: Clock Power Management Test 1 – Required Test .....	199
3.24.2. Test 24b: Clock Power Management Test 2 – Required Test .....	199
<b>3.25. Test 25: Resizable BAR Test – Required Test .....</b>	<b>200</b>
<b>3.26. Test 26: OBFF Enable Test – Required Test .....</b>	<b>200</b>
<b>3.27. Test 27: Various Extended Capabilities Present Test – Required Test .....</b>	<b>201</b>



<b>3.28. Test 28: Various IOV Capabilities Present Test – Required Test .....</b>	<b>201</b>
<b>3.29. Test 29: ARI Configuration Test – Required Test .....</b>	<b>202</b>

<b>APPENDIX A.1. Type 0 Configuration Header AND CAPABILITY .....</b>	<b>203</b>
---	------------

<b>A.1.1. Standard Upstream Port Device Type 0 FUNCTIONS IN TEST CASES .....</b>	<b>203</b>
--	------------

A.1.1.1	PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) – DWORD .....	203
A.1.1.2	PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	203
A.1.1.3	PCI STATUS REGISTER (OFFSET 06h) - WORD .....	204
A.1.1.4	PCI REVISION ID REGISTER (OFFSET 08h) - BYTE .....	204
A.1.1.5	PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES .....	204
A.1.1.6	PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE .....	204
A.1.1.7	PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE .....	205
A.1.1.8	PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE .....	205
A.1.1.9	PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	205
A.1.1.10	PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h) .....	205
A.1.1.11	Default IO BAR – DWORD .....	205
A.1.1.12	Default 32 bit Memory BAR – DWORD .....	205
A.1.1.13	Default 64 bit Memory BAR – 2 DWORDs .....	206
A.1.1.14	PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD .....	206
A.1.1.15	PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD .....	206
A.1.1.16	PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD .....	206
A.1.1.17	PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD .....	206
A.1.1.18	PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	207
A.1.1.19	PCI RESERVED REGISTER (OFFSET 38h) - DWORD .....	207
A.1.1.20	PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	207
A.1.1.21	PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE .....	207
A.1.1.22	PCI MIN_GNT/MAX_LAT REGISTERS (OFFSET 3Eh) - WORD .....	207

<b>A.2. Type 1 Configuration Header .....</b>	<b>208</b>
---	------------

A.2.1. Standard Bridge Device Type 1 Configuration Header .....	208
A.2.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD .....	208
A.2.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	208
A.2.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD .....	208
A.2.1.4 PCI REVISION ID REGISTER (Offset 08h) - BYTE .....	209
A.2.1.5 PCI CLASS CODE REGISTER (Offset 09h) – 3 BYTES .....	209

A.2.1.6	PCI CACHE LINE SIZE REGISTER (Offset 0Ch) – BYTE .....	209
A.2.1.7	PCI MASTER LATENCY TIMER REGISTER (Offset 0Dh) – BYTE .....	209
A.2.1.8	PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE.....	210
A.2.1.9	PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	210
A.2.1.10	PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h).....	210
A.2.1.10.1.	Default IO BAR - DWORD.....	210
A.2.1.10.2.	Default 32 bit Memory BAR - DWORD .....	211
A.2.1.10.3.	Default 64 bit Memory BAR – 2 DWORDs .....	211
A.2.1.11	PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE .....	211
A.2.1.12	PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE.....	211
A.2.1.13	PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE ..	211
A.2.1.14	PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE.	212
A.2.1.15	PCI I/O BASE REGISTER.....	212
A.2.1.15.1.	I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE .....	212
A.2.1.15.2.	I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE .....	212
A.2.1.16	PCI I/O Limit REGISTER.....	212
A.2.1.16.1.	I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE .....	212
A.2.1.16.2.	I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE .....	212
A.2.1.17	PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD .....	213
A.2.1.18	PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD.....	213
A.2.1.19	PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD .....	213
A.2.1.20	PCI PREFETCHABLE MEMORY BASE .....	213
A.2.1.20.1.	PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD .....	213
A.2.1.20.2.	PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD .....	214
A.2.1.21	PCI PREFETCHABLE MEMORY LIMIT .....	214
A.2.1.21.1.	PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD .....	214
A.2.1.21.2.	PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD .....	214
A.2.1.22	PCI PREFETCHABLE MEMORY BASE UPPER 32 .....	214
A.2.1.22.1.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD .....	214
A.2.1.22.2.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD .....	214
A.2.1.23	PCI PREFETCHABLE MEMORY LIMIT UPPER 32.....	215
A.2.1.23.1.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD.....	215
A.2.1.23.2.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD.....	215

A.2.1.24	PCI I/O BASE UPPER 16.....	215
A.2.1.24.1.	I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD.....	215
A.2.1.24.2.	I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD.....	215
A.2.1.25	PCI I/O LIMIT UPPER 16 .....	215
A.2.1.25.1.	I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD.....	215
A.2.1.25.2.	I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD.....	216
A.2.1.26	PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	216
A.2.1.27	PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD .....	216
A.2.1.28	PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	216
A.2.1.29	PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE .....	216
A.2.1.30	PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD.....	217
A.3.	Power Management Capability.....	217
A.3.1.	Standard Type 0 Header Device Power Management Capability Structure	217
A.3.1.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	217
A.3.1.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	218
A.3.1.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	218
A.3.1.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	218
A.3.1.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	218
A.3.2.	Standard Type 1 Header Device Power Management Capability Structure .....	219
A.3.2.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER FIELDS (OFFSET 00h) - WORD.....	219
A.3.2.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	219
A.3.2.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	219
A.3.2.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	220
A.3.2.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	220
<b>A.4.</b>	<b>MSI Capability Structures .....</b>	<b>220</b>
A.4.1.	Standard Device PCI MSI-32 CAPABILITY STRUCTURE.....	220
A.4.1.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	220
A.4.1.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	220
A.4.1.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	221

A.4.1.4	MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD.....	221
A.4.2.	Standard Device PCI MSI-64 CAPABILITY STRUCTURE.....	221
A.4.2.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	221
A.4.2.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	221
A.4.2.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	222
A.4.2.4	MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD..	222
A.4.2.5	MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD .....	222
<b>A.5.</b>	<b>PCI Express Capability Structure .....</b>	<b>222</b>
A.5.1.	Standard Upstream Port Device PCI Express Capability Structure .....	222
A.5.1.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	222
A.5.1.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	223
A.5.1.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	223
A.5.1.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	224
A.5.1.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	224
A.5.1.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	225
A.5.1.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD.....	225
A.5.1.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	225
A.5.2.	Standard Downstream Port Device PCI Express Capability Structure .....	226
A.5.2.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	226
A.5.2.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	226
A.5.2.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	227
A.5.2.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	228
A.5.2.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	228
A.5.2.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	229
A.5.2.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD.....	229
A.5.2.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	230
A.5.3.	Standard Downstream Port Device with Slot PCI Express Capability Structure	230
A.5.3.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	230
A.5.3.2	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	231
A.5.3.3	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	231
A.5.3.4	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	232
A.5.4.	Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure .....	232
A.5.4.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	232
A.5.4.2	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	233
A.5.4.3	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	233
A.5.4.4	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	234

<b>A.6. PCI Express Advanced Error Reporting Capability.....</b>	<b>234</b>
A.6.1. Standard PCI Express Advanced Error Reporting Capability .....	234
A.6.1.1 PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD .....	234
A.6.1.2 UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD	235
A.6.1.3 UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD	236
A.6.1.4 UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD .....	237
A.6.1.5 CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD	237
A.6.1.6 CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD...	238
A.6.1.7 ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD.....	238
A.6.1.8 HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs .....	238
<b>A.7. PCI Express Virtual Channel Capability .....</b>	<b>239</b>
A.7.1. Standard Endpoint PCI Express Virtual Channel Capability.....	239
A.7.1.1 PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	239
A.7.1.2 PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD .....	239
A.7.1.3 PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD .....	239
A.7.1.4 PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD .....	240
A.7.1.5 PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD.....	240
A.7.1.6 VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD.....	240
A.7.1.7 VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD.....	241
A.7.1.8 Reserved Register (OFFSET 18h) - WORD.....	241
A.7.1.9 VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD.....	241
A.7.1.10 VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD ....	241
A.7.1.11 VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD.....	242
A.7.1.12 Reserved Register (OFFSET 24h) - WORD.....	242
A.7.1.13 VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD.....	242
A.7.1.14 VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD.....	243
A.7.1.15 VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD .....	243
A.7.1.16 Reserved Register (OFFSET 30h) - WORD.....	243
A.7.1.17 VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD.....	244
A.7.1.18 VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD.....	244
A.7.1.19 VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD .....	244
A.7.1.20 Reserved Register (OFFSET 3Ch) - WORD .....	245
A.7.1.21 VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD.....	245
A.7.1.22 VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD.....	245
A.7.1.23 VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD.....	246

A.7.1.24	Reserved Register (OFFSET 48h) - WORD.....	246
A.7.1.25	VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD.....	246
A.7.1.26	VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD ....	247
A.7.1.27	VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD .....	247
A.7.1.28	Reserved Register (OFFSET 54h) - WORD.....	247
A.7.1.29	VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD.....	248
A.7.1.30	VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD.....	248
A.7.1.31	VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD .....	248
A.7.1.32	Reserved Register (OFFSET 60h) - WORD.....	249
A.7.1.33	VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD.....	249
A.7.1.34	VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD.....	249
A.7.1.35	VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD .....	250
A.7.1.36	Reserved Register (OFFSET 6Ch) - WORD .....	250
A.7.1.37	VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD.....	250
A.7.1.38	VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs .....	250
<b>A.8.</b>	<b>PCI Express Power Budgeting Capability .....</b>	<b>251</b>
A.8.1.	Standard Device PCI Express Power Budgeting Capability .....	251
A.8.1.1	PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	251
A.8.1.2	DATA SELECT REGISTER (OFFSET 04h) - DWORD .....	251
A.8.1.3	DATA REGISTER (OFFSET 08h) - DWORD .....	251
A.8.1.4	POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD ....	252
<b>APPENDIX B.</b>	<b>PCI EXPRESS 2.0 REGISTER AND CAPABILITY – DEFAULTS, CONTENTS, AND CHARACTERISTICS FOR FUNCTIONS IN TEST CASES .....</b>	<b>253</b>
<b>B.1.</b>	<b>Type 0 Configuration Header .....</b>	<b>253</b>
B.1.1.	Standard Upstream Port Type 0 Configuration Header .....	253
B.1.1.1	PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD.....	253
B.1.1.2	PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	254
B.1.1.3	PCI STATUS REGISTER (OFFSET 06h) - WORD .....	254
B.1.1.4	PCI REVISION ID REGISTER (OFFSET 08h) - BYTE.....	254
B.1.1.5	PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES.....	255
B.1.1.6	PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE .....	255
B.1.1.7	PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE .....	255
B.1.1.8	PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE.....	255
B.1.1.9	PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	255
B.1.1.10	PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h).....	256
B.1.1.10.1.	Default IO BAR - DWORD.....	256

B.1.1.10.2. Default 32 bit Memory BAR - DWORD .....	256
B.1.1.10.3. Default 64 bit Memory BAR – 2 DWORDs .....	256
B.1.1.11 PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD .....	256
B.1.1.12 PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD.....	256
B.1.1.13 PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD .....	257
B.1.1.14 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD .....	257
B.1.1.15 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	257
B.1.1.16 PCI RESERVED REGISTER (OFFSET 38h) - DWORD .....	257
B.1.1.17 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	257
B.1.1.18 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE .....	258
B.1.1.19 PCI MIN_GNT/MAX_LAT REGISTERS (OFFSET 3Eh) - WORD .....	258
<b>B.2. Type 1 Configuration Header .....</b>	<b>258</b>
B.2.1. Standard Bridge Device Type 1 Configuration Header.....	258
B.2.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD.....	258
B.2.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	259
B.2.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD .....	259
B.2.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE.....	259
B.2.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES.....	260
B.2.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE .....	260
B.2.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE .....	260
B.2.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE.....	260
B.2.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	260
B.2.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h).....	261
B.2.1.10.1. Default IO BAR - DWORD.....	261
B.2.1.10.2. Default 32 bit Memory BAR - DWORD .....	261
B.2.1.10.3. Default 64 bit Memory BAR – 2 DWORDs .....	261
B.2.1.11 PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE .....	261
B.2.1.12 PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE.....	262
B.2.1.13 PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE ..	262
B.2.1.14 PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE.	262
B.2.1.15 PCI I/O BASE REGISTER.....	262
B.2.1.15.1. I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE .....	262
B.2.1.15.2. I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE .....	262
B.2.1.16 PCI I/O Limit REGISTER.....	263
B.2.1.16.1. I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE .....	263
B.2.1.16.2. I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE .....	263
B.2.1.17 PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD .....	263
B.2.1.18 PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD.....	263

B.2.1.19	PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD .....	264
B.2.1.20	PCI PREFETCHABLE MEMORY BASE .....	264
B.2.1.20.1.	PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD .....	264
B.2.1.20.2.	PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD .....	264
B.2.1.21	PCI PREFETCHABLE MEMORY LIMIT .....	264
B.2.1.21.1.	PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD .....	264
B.2.1.21.2.	PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD .....	265
B.2.1.22	PCI PREFETCHABLE MEMORY BASE UPPER 32 .....	265
B.2.1.22.1.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD .....	265
B.2.1.22.2.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD .....	265
B.2.1.23	PCI PREFETCHABLE MEMORY LIMIT UPPER 32.....	265
B.2.1.23.1.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD.....	265
B.2.1.23.2.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD.....	265
B.2.1.24	PCI I/O BASE UPPER 16.....	266
B.2.1.24.1.	I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD .....	266
B.2.1.24.2.	I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD .....	266
B.2.1.25	PCI I/O LIMIT UPPER 16 .....	266
B.2.1.25.1.	I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD .....	266
B.2.1.25.2.	I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD .....	266
B.2.1.26	PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	266
B.2.1.27	PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD .....	267
B.2.1.28	PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	267
B.2.1.29	PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE .....	267
B.2.1.30	PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD.....	267
<b>B.3.</b>	<b>Power Management Capability.....</b>	<b>268</b>
B.3.1.	Standard Type 0 Header Device Power Management Capability Structure .....	268
B.3.1.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	268
B.3.1.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	268



B.3.1.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	268
B.3.1.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	269
B.3.1.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	269
B.3.2.	Standard Type 1 Header Device Power Management Capability Structure .....	269
B.3.2.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	269
B.3.2.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	269
B.3.2.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	270
B.3.2.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	270
B.3.2.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	270
<b>B.4.</b>	<b>MSI Capability Structures .....</b>	<b>271</b>
B.4.1.	Standard Device PCI MSI-32 CAPABILITY STRUCTURE.....	271
B.4.1.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	271
B.4.1.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	271
B.4.1.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	271
B.4.1.4	MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD.....	272
B.4.2.	Standard Device PCI MSI-64 CAPABILITY STRUCTURE.....	272
B.4.2.1	MSI CAPABILITY FIELDS (OFFSET 00h) .....	272
B.4.2.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	272
B.4.2.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	272
B.4.2.4	MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD..	273
B.4.2.5	MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD .....	273
B.4.2.6	Standard Device PCI MSI-PVM-32 CAPABILITY STRUCTURE .....	273
B.4.2.7	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	273
B.4.2.8	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	273
B.4.2.9	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	274
B.4.2.10	MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD.....	274
B.4.2.11	MSI RESERVED REGISTER (OFFSET 0Ah) - WORD.....	274
B.4.2.12	MSI MASK BITS REGISTER (OFFSET 0Ch) - DWORD.....	274
B.4.2.13	MSI PENDING BITS REGISTER (OFFSET 10h) - DWORD .....	274
B.4.3.	Standard Device PCI MSI-PVM-64 CAPABILITY STRUCTURE.....	275
B.4.3.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	275
B.4.3.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	275
B.4.3.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	275

B.4.3.4	MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD .....	276
B.4.3.5	MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD .....	276
B.4.3.6	MSI RESERVED REGISTER (OFFSET 0Eh) - WORD .....	276
B.4.3.7	MSI MASK BITS REGISTER (OFFSET 10h) - DWORD .....	276
B.4.3.8	MSI PENDING BITS REGISTER (OFFSET 14h) - DWORD .....	276
<b>B.5.</b>	<b>PCI Express Capability Structure .....</b>	<b>277</b>
B.5.1.	Standard Upstream Port Device PCI Express Capability Structure .....	277
B.5.1.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	277
B.5.1.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	277
B.5.1.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	278
B.5.1.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	278
B.5.1.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	279
B.5.1.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	279
B.5.1.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD .....	280
B.5.1.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	280
B.5.1.9	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	280
B.5.1.10	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	281
B.5.1.11	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	281
B.5.1.12	ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD .....	281
B.5.1.13	ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD .....	281
B.5.1.14	ROOT STATUS REGISTER (OFFSET 20h) - DWORD .....	281
B.5.1.15	DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD .....	281
B.5.1.16	DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD .....	282
B.5.1.17	DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD .....	282
B.5.1.18	LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD .....	282
B.5.1.19	LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD .....	282
B.5.1.20	LINK STATUS 2 REGISTER (OFFSET 32h) - WORD .....	283
B.5.1.21	SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD .....	283
B.5.1.22	SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD .....	283
B.5.1.23	SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD .....	283
B.5.2.	Standard Downstream Port Device PCI Express Capability Structure .....	284
B.5.2.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	284
B.5.2.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	284
B.5.2.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	285
B.5.2.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	285
B.5.2.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	286

B.5.2.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	286
B.5.2.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD .....	287
B.5.2.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	287
B.5.2.9	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	287
B.5.2.10	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	288
B.5.2.11	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	288
B.5.2.12	ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD .....	288
B.5.2.13	ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD .....	288
B.5.2.14	ROOT STATUS REGISTER (OFFSET 20h) - DWORD .....	288
B.5.2.15	DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD .....	289
B.5.2.16	DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD .....	289
B.5.2.17	DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD .....	289
B.5.2.18	LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD .....	289
B.5.2.19	LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD .....	290
B.5.2.20	LINK STATUS 2 REGISTER (OFFSET 32h) - WORD .....	290
B.5.2.21	SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD .....	290
B.5.2.22	SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD .....	290
B.5.2.23	SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD .....	291
B.5.3.	Standard Downstream Port Device with Slot PCI Express Capability Structure .....	291
B.5.3.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	291
B.5.3.2	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	291
B.5.3.3	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	292
B.5.3.4	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	292
B.5.4.	Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure .....	293
B.5.4.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	293
B.5.4.2	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	293
B.5.4.3	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	293
B.5.4.4	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	293
B.5.4.5	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	294
B.5.4.6	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	294
<b>B.6.</b>	<b>PCI Express Advanced Error Reporting Capability .....</b>	<b>295</b>
B.6.1.	Standard PCI Express Advanced Error Reporting Capability .....	295
B.6.1.1	PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD .....	295
B.6.1.2	UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD	296
B.6.1.3	UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD	297

B.6.1.4	UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD .....	298
B.6.1.5	CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD	298
B.6.1.6	CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD....	299
B.6.1.7	ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD.....	299
B.6.1.8	HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs .....	299
<b>B.7.</b>	<b>PCI Express Virtual Channel Capability .....</b>	<b>300</b>
B.7.1.	Standard Endpoint PCI Express Virtual Channel Capability .....	300
B.7.1.1	PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	300
B.7.1.2	PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD .....	300
B.7.1.3	PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD .....	300
B.7.1.4	PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD .....	301
B.7.1.5	PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD.....	301
B.7.1.6	VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD .....	301
B.7.1.7	VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD .....	302
B.7.1.8	Reserved Register (OFFSET 18h) - WORD.....	302
B.7.1.9	VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD.....	302
B.7.1.10	VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD .....	303
B.7.1.11	VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD .....	303
B.7.1.12	Reserved Register (OFFSET 24h) - WORD.....	303
B.7.1.13	VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD .....	304
B.7.1.14	VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD....	304
B.7.1.15	VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD .....	304
B.7.1.16	Reserved Register (OFFSET 30h) - WORD.....	305
B.7.1.17	VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD.....	305
B.7.1.18	VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD....	305
B.7.1.19	VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD .....	306
B.7.1.20	Reserved Register (OFFSET 3Ch) - WORD .....	306
B.7.1.21	VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD.....	306
B.7.1.22	VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD.....	307
B.7.1.23	VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD .....	307
B.7.1.24	Reserved Register (OFFSET 48h) - WORD.....	307
B.7.1.25	VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD.....	308

B.7.1.26	VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD ....	308
B.7.1.27	VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD .....	308
B.7.1.28	Reserved Register (OFFSET 54h) - WORD .....	309
B.7.1.29	VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD .....	309
B.7.1.30	VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD .....	309
B.7.1.31	VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD .....	310
B.7.1.32	Reserved Register (OFFSET 60h) - WORD .....	310
B.7.1.33	VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD .....	310
B.7.1.34	VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD .....	311
B.7.1.35	VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD .....	311
B.7.1.36	Reserved Register (OFFSET 6Ch) - WORD .....	311
B.7.1.37	VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD .....	312
B.7.1.38	VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs .....	312
<b>B.8.</b>	<b>PCI Express Power Budgeting Capability .....</b>	<b>312</b>
B.8.1.	Standard Device PCI Express Power Budgeting Capability .....	312
B.8.1.1	PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	312
B.8.1.2	DATA SELECT REGISTER (OFFSET 04h) - DWORD .....	312
B.8.1.3	DATA REGISTER (OFFSET 08h) - DWORD .....	313
B.8.1.4	POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD ....	313
<b>APPENDIX C.</b>	<b>PCI EXPRESS 3.0 REGISTER AND CAPABILITY – DEFAULTS, CONTENTS, AND CHARACTERISTICS FOR FUNCTIONS IN TEST CASES .....</b>	<b>314</b>
<b>C.1.</b>	<b>Type 0 Configuration Header .....</b>	<b>314</b>
C.1.1.	Standard Upstream Port Type 0 Configuration Header .....	314
C.1.1.1	PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD .....	314
C.1.1.2	PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	314
C.1.1.3	PCI STATUS REGISTER (OFFSET 06h) - WORD .....	315
C.1.1.4	PCI REVISION ID REGISTER (OFFSET 08h) - BYTE .....	315
C.1.1.5	PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES .....	316
C.1.1.6	PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE .....	316
C.1.1.7	PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE .....	316
C.1.1.8	PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE .....	316
C.1.1.9	PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	316
C.1.1.10	PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h) .....	317
C.1.1.10.1.	Default IO BAR - DWORD .....	317
C.1.1.10.2.	Default 32 bit Memory BAR - DWORD .....	317
C.1.1.10.3.	Default 64 bit Memory BAR – 2 DWORDs .....	317

C.1.1.11	PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD .....	317
C.1.1.12	PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD.....	318
C.1.1.13	PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD .....	318
C.1.1.14	PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD .....	318
C.1.1.15	PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	318
C.1.1.16	PCI RESERVED REGISTER (OFFSET 38h) - DWORD .....	318
C.1.1.17	PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	319
C.1.1.18	PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE .....	319
C.1.1.19	PCI MIN_GNT/MAX_LAT REGISTERS (OFFSET 3Eh) - WORD .....	319
C.2.	Type 1 Configuration Header .....	319
C.2.1.	Standard Bridge Device Type 1 Configuration Header.....	319
C.2.1.1	PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD.....	319
C.2.1.2	PCI COMMAND REGISTER (OFFSET 04h) - WORD .....	320
C.2.1.3	PCI STATUS REGISTER (OFFSET 06h) - WORD .....	320
C.2.1.4	PCI REVISION ID REGISTER (OFFSET 08h) - BYTE.....	321
C.2.1.5	PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES.....	321
C.2.1.6	PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE .....	321
C.2.1.7	PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE .....	321
C.2.1.8	PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE.....	321
C.2.1.9	PCI BIST REGISTER (OFFSET 0Fh) - BYTE .....	322
C.2.1.10	PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h).....	322
C.2.1.10.1.	Default IO BAR - DWORD .....	322
C.2.1.10.2.	Default 32 bit Memory BAR - DWORD .....	322
C.2.1.10.3.	Default 64 bit Memory BAR – 2 DWORDs .....	322
C.2.1.11	PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE .....	323
C.2.1.12	PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE.....	323
C.2.1.13	PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE ..	323
C.2.1.14	PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE.....	323
C.2.1.15	PCI I/O BASE REGISTER.....	323
C.2.1.15.1.	I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE .....	323
C.2.1.15.2.	I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE .....	324
C.2.1.16	PCI I/O Limit REGISTER.....	324
C.2.1.16.1.	I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE .....	324
C.2.1.16.2.	I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE .....	324
C.2.1.17	PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD .....	324

C.2.1.18	PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD.....	325
C.2.1.19	PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD .....	325
C.2.1.20	PCI PREFETCHABLE MEMORY BASE .....	325
C.2.1.20.1.	PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD .....	325
C.2.1.20.2.	PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD .....	325
C.2.1.21	PCI PREFETCHABLE MEMORY LIMIT .....	326
C.2.1.21.1.	PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD .....	326
C.2.1.21.2.	PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD .....	326
C.2.1.22	PCI PREFETCHABLE MEMORY BASE UPPER 32 .....	326
C.2.1.22.1.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD .....	326
C.2.1.22.2.	PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD .....	326
C.2.1.23	PCI PREFETCHABLE MEMORY LIMIT UPPER 32.....	326
C.2.1.23.1.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD.....	326
C.2.1.23.2.	PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD.....	327
C.2.1.24	PCI I/O BASE UPPER 16.....	327
C.2.1.24.1.	I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD .....	327
C.2.1.24.2.	I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD .....	327
C.2.1.25	PCI I/O LIMIT UPPER 16 .....	327
C.2.1.25.1.	I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD .....	327
C.2.1.25.2.	I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD .....	327
C.2.1.26	PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD .....	328
C.2.1.27	PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD .....	328
C.2.1.28	PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE .....	328
C.2.1.29	PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) – BYTE .....	328
C.2.1.30	PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD.....	329
<b>C.3.</b>	<b>Power Management Capability.....</b>	<b>329</b>
C.3.1.	Standard Type 0 Header Device Power Management Capability Structure .....	329
C.3.1.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD.....	329

C.3.1.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	330
C.3.1.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	330
C.3.1.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	330
C.3.1.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	331
C.3.2.	Standard Type 1 Header Device Power Management Capability Structure	331
C.3.2.1	PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	331
C.3.2.2	POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD.....	331
C.3.2.3	POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD .....	332
C.3.2.4	PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR_BSE) (OFFSET 06h) - BYTE .....	332
C.3.2.5	POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE.....	332
<b>C.4.</b>	<b>MSI Capability Structures .....</b>	<b>333</b>
C.4.1.	Standard Device PCI MSI-32 CAPABILITY STRUCTURE.....	333
C.4.1.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	333
C.4.1.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	333
C.4.1.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	333
C.4.1.4	MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD.....	333
C.4.2.	Standard Device PCI MSI-64 CAPABILITY STRUCTURE.....	334
C.4.2.1	MSI CAPABILITY FIELDS (OFFSET 00h) .....	334
C.4.2.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	334
C.4.2.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	334
C.4.2.4	MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD..	334
C.4.2.5	MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD.....	335
C.4.3.	Standard Device PCI MSI-PVM-32 CAPABILITY STRUCTURE.....	335
C.4.3.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	335
C.4.3.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	335
C.4.3.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	335
C.4.3.4	MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD.....	336
C.4.3.5	MSI RESERVED REGISTER (OFFSET 0Ah) - WORD.....	336
C.4.3.6	MSI MASK BITS REGISTER (OFFSET 0Ch) - DWORD.....	336
C.4.3.7	MSI PENDING BITS REGISTER (OFFSET 10h) - DWORD .....	336
C.4.4.	Standard Device PCI MSI-PVM-64 CAPABILITY STRUCTURE.....	336
C.4.4.1	MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD	336
C.4.4.2	MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD.....	337



C.4.4.3	MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD .....	337
C.4.4.4	MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD..	337
C.4.4.5	MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD .....	337
C.4.4.6	MSI RESERVED REGISTER (OFFSET 0Eh) - WORD.....	337
C.4.4.7	MSI MASK BITS REGISTER (OFFSET 10h) - DWORD .....	337
C.4.4.8	MSI PENDING BITS REGISTER (OFFSET 14h) - DWORD .....	338
C.5.	PCI Express Capability Structure .....	338
C.5.1.	Standard Upstream Port Device PCI Express Capability Structure .....	338
C.5.1.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	338
C.5.1.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	338
C.5.1.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	339
C.5.1.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	339
C.5.1.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	340
C.5.1.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	340
C.5.1.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD.....	341
C.5.1.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	341
C.5.1.9	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	341
C.5.1.10	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	342
C.5.1.11	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	342
C.5.1.12	ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD.....	342
C.5.1.13	ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD .....	342
C.5.1.14	ROOT STATUS REGISTER (OFFSET 20h) - DWORD .....	342
C.5.1.15	DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD .....	343
C.5.1.16	DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD .....	343
C.5.1.17	DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD .....	343
C.5.1.18	LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD .....	344
C.5.1.19	LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD.....	344
C.5.1.20	LINK STATUS 2 REGISTER (OFFSET 32h) - WORD .....	345
C.5.1.21	SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD .....	345
C.5.1.22	SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD .....	345
C.5.1.23	SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD .....	345
C.5.2.	Standard Downstream Port Device PCI Express Capability Structure .....	346
C.5.2.1	PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD .....	346
C.5.2.2	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD .....	346
C.5.2.3	DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD .....	347
C.5.2.4	DEVICE CONTROL REGISTER (OFFSET 08h) - WORD .....	347
C.5.2.5	DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD .....	348

C.5.2.6	LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD .....	348
C.5.2.7	LINK CONTROL REGISTER (OFFSET 10h) - WORD .....	349
C.5.2.8	LINK STATUS REGISTER (OFFSET 12h) - WORD .....	349
C.5.2.9	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	349
C.5.2.10	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	350
C.5.2.11	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	350
C.5.2.12	ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD .....	350
C.5.2.13	ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD .....	350
C.5.2.14	ROOT STATUS REGISTER (OFFSET 20h) - DWORD .....	350
C.5.2.15	DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD .....	351
C.5.2.16	DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD .....	351
C.5.2.17	DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD .....	352
C.5.2.18	LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD .....	352
C.5.2.19	LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD .....	352
C.5.2.20	LINK STATUS 2 REGISTER (OFFSET 32h) - WORD .....	353
C.5.2.21	SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD .....	353
C.5.2.22	SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD .....	353
C.5.2.23	SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD .....	353
C.5.3.	Standard Downstream Port Device with Slot PCI Express Capability Structure .....	354
C.5.3.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	354
C.5.3.2	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	354
C.5.3.3	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	355
C.5.3.4	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	355
C.5.4.	Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure .....	356
C.5.4.1	PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD .....	356
C.5.4.2	SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD .....	356
C.5.4.3	SLOT CONTROL REGISTER (OFFSET 18h) - WORD .....	357
C.5.4.4	SLOT STATUS REGISTER (OFFSET 1Ah) - WORD .....	357
<b>C.6.</b>	<b>PCI Express Advanced Error Reporting Capability .....</b>	<b>358</b>
C.6.1.	Standard PCI Express Advanced Error Reporting Capability (without TLP Prefix Support) .....	358
C.6.1.1	PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD .....	358
C.6.1.2	UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD .....	359
C.6.1.3	UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD .....	360
C.6.1.4	UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD .....	361

C.6.1.5	CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD .....	362
C.6.1.6	CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD .....	362
C.6.1.7	ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD .....	363
C.6.1.8	HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs .....	363
C.6.2.	Standard PCI Express Advanced Error Reporting Capability with TLP Prefix Support .....	363
C.6.2.1	PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD .....	363
C.6.2.2	ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD .....	364
C.6.2.3	ROOT ERROR COMMAND REGISTER (OFFSET 2Ch) –DWORD .....	364
C.6.2.4	ROOT ERROR STATUS REGISTER (OFFSET 30h) –DWORD .....	364
C.6.2.5	CORRECTABLE ERROR SOURCE IDENTIFICATION REGISTER (OFFSET 34h) –WORD .....	364
C.6.2.6	ERROR SOURCE IDENTIFICATION REGISTER (OFFSET 36h) –WORD .....	364
C.6.2.7	TLP PREFIX LOG REGISTER (OFFSET 38h) – 4 DWORDs .....	365
<b>C.7.</b>	<b>PCI Express Virtual Channel Capability .....</b>	<b>365</b>
C.7.1.	Standard Endpoint PCI Express Virtual Channel Capability .....	365
C.7.1.1	PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	365
C.7.1.2	PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD .....	365
C.7.1.3	PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD .....	366
C.7.1.4	PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD .....	366
C.7.1.5	PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD .....	366
C.7.1.6	VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD .....	366
C.7.1.7	VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD .....	367
C.7.1.8	Reserved Register (OFFSET 18h) - WORD .....	367
C.7.1.9	VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD .....	367
C.7.1.10	VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD .....	367
C.7.1.11	VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD .....	368
C.7.1.12	Reserved Register (OFFSET 24h) - WORD .....	368
C.7.1.13	VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD .....	368
C.7.1.14	VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD .....	369
C.7.1.15	VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD .....	369
C.7.1.16	Reserved Register (Offset 30h) - WORD .....	369
C.7.1.17	VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD .....	370
C.7.1.18	VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD .....	370

C.7.1.19	VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD .....	371
C.7.1.20	Reserved Register (OFFSET 3Ch) - WORD .....	371
C.7.1.21	VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD.....	371
C.7.1.22	VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD.....	372
C.7.1.23	VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD.....	372
C.7.1.24	Reserved Register (OFFSET 48h) - WORD.....	372
C.7.1.25	VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD.....	373
C.7.1.26	VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD ....	373
C.7.1.27	VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD.....	373
C.7.1.28	Reserved Register (OFFSET 54h) - WORD.....	374
C.7.1.29	VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD.....	374
C.7.1.30	VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD.....	374
C.7.1.31	VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD .....	375
C.7.1.32	Reserved Register (OFFSET 60h) - WORD.....	375
C.7.1.33	VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD.....	375
C.7.1.34	VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD.....	376
C.7.1.35	VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD.....	376
C.7.1.36	Reserved Register (OFFSET 6Ch) - WORD .....	376
C.7.1.37	VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD.....	377
C.7.1.38	VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs .....	377
<b>C.8.</b>	<b>PCI Express Power Budgeting Capability .....</b>	<b>377</b>
C.8.1.	Standard Device PCI Express Power Budgeting Capability .....	377
C.8.1.1	PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD .....	377
C.8.1.2	DATA SELECT REGISTER (OFFSET 04h) - DWORD .....	377
C.8.1.3	DATA REGISTER (OFFSET 08h) - DWORD .....	378
C.8.1.4	POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD ....	378
<b>C.9.</b>	<b>PCI Express Secondary PCI Express Extended Capability .....</b>	<b>379</b>
C.9.1.	Standard Upstream Port PCI Express Secondary PCI Express Extended Capability (without Crosslink Support) .....	379
C.9.1.1	PCI EXPRESS SECONDARY PCI EXPRESS EXTENDED CAPABILITY HEADER (OFFSET 00h) - DWORD.....	379
C.9.1.2	LINK CONTROL 3 REGISTER (OFFSET 04h) - DWORD .....	379
C.9.1.3	LANE ERROR STATUS REGISTER (OFFSET 08h) - DWORD .....	379
C.9.1.4	EQUALIZATION CONTROL REGISTER (OFFSET 0Ch) - WORD.....	380
C.9.2.	Standard Downstream Port PCI Express Secondary PCI Express Extended Capability .....	380
C.9.2.1	PCI EXPRESS SECONDARY PCI EXPRESS EXTENDED CAPABILITY HEADER (OFFSET 00h) - DWORD.....	380

C.9.2.2	LINK CONTROL 3 REGISTER (OFFSET 04h) - DWORD .....	380
C.9.2.3	LANE ERROR STATUS REGISTER (OFFSET 08h) - DWORD .....	381
C.9.2.4	EQUALIZATION CONTROL REGISTER (OFFSET 0Ch) - WORD .....	381

<b>APPENDIX D. ACKNOWLEDGEMENTS .....</b>	<b>382</b>
---	------------

# 1

## 1. Introduction

This test specification primarily covers tests of PCI Express platform firmware for features critical to PCI Express. This specification does not include the complete set of tests for a PCI Express System.

In particular, a platform must also meet the requirements and tests described in the latest versions of the following documents as well as any other tests provided by the PCI-SIG:

- ❑ PCI Express Electrical Test Considerations
- ❑ PCI Express Configuration Space Validation Test Considerations
- ❑ PCI Express Link Test Considerations
- ❑ PCI Express Transaction Test Considerations
- ❑ This test specification is written for hardware that is designed to the following specifications:
  - ❑ PCI Express Base Specification, Revision 1.1 – For Appendix A
  - ❑ PCI Express Base Specification, Revision 2.0 – For Appendix B
  - ❑ PCI Express Base Specification, Revision 3.0 – v1.0

This specification provides a list of test descriptions relating to Chapters 5, 6 and 7 of the PCI Express Base Specification required for PCI Express platforms. These descriptions provide a partial list of criteria that systems must meet for PCI Express compliance testing.

## 2. Test Descriptions

### 2.1. Test Topologies and Hardware Functionality

Many of the tests described in this specification require the use of a PCI Express Test Device with platform test capabilities. The device must provide a programmable means to produce interrupts, errors, PME events, active state power management transitions, and to simulate faulty device register implementations. One or more of the platform test devices can be used to represent complex PCI Express topologies including multiple PCI Express bridges and complex memory assignments. It is not the intent of this specification to cover the exact implementation or programming details of such a test device. The specification specifies the exact requirements for test hardware behavior at the configuration/protocol levels. There are a variety of options for hardware implementations to meet these requirements.

For platforms with slots that support greater than 2.5 GT/s link speeds, tests should be run at all supported link speeds.

#### 2.1.1. Platform Init/Config Tests

The following tests perform basic high level functionality checks of a PCI Express capable system. The tests focus on areas where system firmware could make mistakes in supporting features new to PCI Express or general complex PCI topologies.

#### 2.1.2. General Topology Test Checks

This section describes checks performed by all tests where a test topology is being enumerated by the system under test. All topology tests perform these checks. The tests are only described in this section. Tests that perform these checks refer back to this section.

### 2.1.2.1. Memory BAR Configuration

All memory BARs must be configured with non-overlapping addresses. Memory Space Enable must be turned on for all functions requesting memory. The test software walks all functions and all memory BARs after configuration. The basic algorithm performed by the test software is described below. Any additional checks that are performed are described in individual test cases.

1. The test software prompts the user to specify the amount of free memory in the system before the test case begins.
2. For each PCI Express function in the system, the test software reads the contents of the Command register at offset 04h in configuration space after the configuration process is complete.
3. If all of a function's memory BARs could be enabled without exceeding the available system memory the Memory Space Enable bit in the Command register for that function must be one when the system firmware finishes executing.
4. Test software tracks that all memory BARs are sized during the configuration process. The system must attempt to write 1 to all readable bits in the BAR and then read the BAR. Failure to do this for any BAR for any function results in a test failure.
5. Test software tracks and records the size of all implemented memory BAR registers by monitoring the size read from the BARs during the configuration process.
6. Test software reads all memory BAR register contents for all functions in the test topology once the configuration process has finished.
7. Using the base address and size information for all implemented registers test software checks to see if any of the assigned memory ranges overlap. Any overlap is recorded as a test failure. When checking for overlaps, expansion ROM sizes and locations are also considered. More detailed information on standard tests for Expansion ROM cases are covered in Section 2.1.2.3 on Expansion ROM configuration.
8. Test software monitors accesses performed by the system throughout the configuration process. If any BAR (even if it is zero size or not implemented) is not accessed by the system to determine size, the test reports a failure.
- 0.



### 2.1.2.2. IO BAR Configuration

All IO BARs must be configured with non-overlapping addresses. I/O Space Enable must be turned on for all functions requesting IO. The test software walks all functions and all IO BARs after configuration. The basic algorithm performed by the test software is described below. Any additional checks that are performed are described in individual test cases.

1. The test software prompts the user to specify the amount of free IO in the system before the test case begins.
2. For each PCI Express function in the system, the test software reads the contents of the Command register at offset 04h in configuration space after the configuration process is complete.
3. If all of a function's IO BARs could be enabled without exceeding the available system IO the I/O Space Enable bit in the Command register for that function must be one when the system firmware finishes executing.
4. Test software tracks that all IO BARs are sized during the configuration process. The system must attempt to write 1 to all readable bits in the BAR and then read the BAR. Failure to do this for any BAR for any function results in a test failure.
5. Test software tracks and records the size of all implemented IO BAR registers by monitoring the size read from the BARs during the configuration process.
6. Test software reads all IO BAR register contents for all functions in the test topology once the configuration process has finished.
7. Using the base address and size information for all implemented registers test software checks to see if any of the assigned IO ranges overlap. Any overlap is recorded as a test failure.
8. Test software monitors accesses performed by the system throughout the configuration process. If any BAR (even if it is zero size or not implemented) is not accessed by the system to determine size, the test reports a failure.
- 0.

### 2.1.2.3. Expansion ROM Configuration

A number of test cases contain one or more option ROMs. The following sketches the general algorithm used to test Expansion ROM configuration performed by all test cases with one or more Expansion ROMs. If additional checks relating to Expansion ROM configuration are performed, they are described in individual test cases.

1. Test software determines the type of system under test.
2. Test software monitors the configuration steps performed by the system under test.
3. Test software tracks that all Expansion ROM BARs are sized during the configuration process. The system must attempt to write 1 to all readable bits in the BAR and then read the BAR. Failure to do this for any Expansion ROM BAR for any function results in a test failure.

4. If there is available memory for any Expansion ROM to be mapped to system memory space this mapping must be performed to an unoccupied location. Test software monitors the locations assigned to all Expansion ROMs and verifies that they do not overlap with other memory ranges assigned to functions in the test topology. If space is available and an Expansion ROM BAR is not assigned an address and enabled the test fails.
5. Test software monitors that the system checks the signatures of the one of more ROM images in the Expansion ROM. The test fails if there is not a compatible image in the ROM, but the system fails to check the signatures of all images.
6. If a compatible image is present with a Vendor ID and Device ID that match the values in the configuration space of the function, test software monitors to ensure that the entire image is read by the system under test.



**Note:** Steps 7 - 9 apply only for PC compatible systems/

7. Test software monitors to ensure that the Expansion ROM Enable bit is zero.
  8. Test software monitors system behavior to verify that the ROM image is executed. The ROM images are programmed to perform a specific operation that is visible to the test software. The operation is specific to the test hardware implementation and is not specified in this specification.
  9. In a few test cases the appropriate ROM image for the system under test does not have a Vendor ID and Device ID signature that matches those found in configuration space. It is a failure if the system executes the ROM image anyway in these cases.
  10. In a few test cases the appropriate ROM image for the system under test does not have a valid checksum. It is a failure if the system executes the ROM image anyway in these cases.
- 0.

### 2.1.2.4. Bridge Address Assignment

Bridge assignment within the test topology of devices must be unique. The following steps are performed by test software to check bus assignment for any test case involving one or more type 1 header component.

1. The test software reads the values written by system configuration software to each of the following registers for all type 1 header devices in the test topology.
    - Primary Bus Number
    - Secondary Bus Number
    - Subordinate Bus Number
  2. Test software compares the contents of the Secondary Bus Number register of the parent with the Primary Bus Number register of any immediate children that are type 1 header devices. The values must match or the test fails.
  3. Test software checks the bus numbers assigned to all bus segments in the test topology and ensures that they are unique.
  4. For every type 1 header device in the test topology the test software computes the bus numbers assigned to each bus segment on the secondary side of the type 1 device. If any of these bus segment numbers do not exceed the Primary Bus Number register value, the test fails. If any of these bus segment numbers exceed the Subordinate Bus Number register value the test fails. If the highest number assigned to a bus segment behind the type 1 header device is not equal to the Subordinate Bus Number register the test fails.
- 0.

### 2.1.2.5. Bridge Memory and IO Window Configuration

1. The test software reads the values written by system configuration software to each of the following registers for all type 1 header devices in the test topology.
  - IO Base
  - IO Base Upper 16 Bits
  - IO Limit
  - IO Limit Upper 16 Bits
  - Memory Base
  - Memory Limit
  - Prefetchable Memory Base
  - Prefetchable Base Upper 32 Bits
  - Prefetchable Memory Limit
  - Prefetchable Limit Upper 32 Bits
2. The base and limit registers for every child type 1 header device are compared with those same registers in its direct parent device. If the window in the child range is not a subset of the parent range or configured so that the limit register value is less than the base register value (ie. range is disabled) the test fails.
3. For each type 1 header device, the resource allocations for all Memory BARs on subordinate components are noted. If any of these resource allocations fall outside of the range specified by the Memory base and limit registers the test fails.
4. For each type 1 header device, the resource allocations for all Prefetchable Memory BARs on subordinate components are noted. If any of these resource allocations fall outside of the range specified by the Prefetchable base and limit registers the test fails.
5. For each type 1 header device, the resource allocations for all IO BARs on subordinate components are noted. If any of these resource allocations fall outside of the range specified by the IO base and limit registers the test fails.
6. For any type 1 header device, if no subordinate component has a memory BAR the Memory Limit register must have been written to a value less than or equal to the Memory Base register or the test fails.
7. For any type 1 header device, if no subordinate component has a prefetchable memory BAR the Prefetchable Memory Limit register must have been written to a value less than or equal to the Prefetchable Memory Base register or the test fails.
8. For any type 1 header device, if no subordinate component has an IO BAR the IO Limit register must have been written to a value less than or equal to the IO Base register or the test fails.

## 2.2. Devices in Test Topologies

Each of the test cases describes a set of bridges and devices. The default register contents for bridges and devices used in test cases are described here. Each individual test case notes any deviations from the default configuration for any device or bridge in that test case.

### 2.2.1. Default registers and capability structures

Appendices A, B and C define default register contents for devices compliant to the 1.1, 2.0 and 3.0 PCI Express Base Specifications.

### 2.2.2. Default PCI Express Devices

Default devices for use in test case descriptions are now defined using the default registers contents and capabilities that can be defined. These default devices are the building blocks used to describe many of the test topologies in this specification. A default device is specified for PCI Express 1.1, 2.0 and 3.0 Base specifications.

#### 2.2.2.1. Default PCI Express Endpoint

A default PCI Express Endpoint has the following default register contents and capabilities:

- ❑ Standard Upstream Port Type 0 Configuration Header

- ❑ BARs

```
Bar1.type = Mem64
Bar1.size = 0
Bar3.type = Mem64
Bar3.size = 0
Bar5.type = Mem64
Bar5.size = 0
```

- ❑ PCI Capabilities

```
[Power Management Capability Structure]
Type = EndPoint
PCI_POWER_MANAGEMENT.present = 1
PCI_POWER_MANAGEMENT.offset = 0x40
PCI_POWER_MANAGEMENT.nextcap = 0x50
[MSI-64 Capability Structure]
Type = EndPoint
MSI_64.present = 1
MSI_64.offset = 0x50
MSI_64.nextcap = 0x60
[PCI Express Capabilities Structure]
Type = EndPoint (Standard Upstream Port PCI Express Capability
Structure)
```

- `PCI_EXPRESS.present = 1`  
`PCI_EXPRESS.offset = 0x60`  
`PCI_EXPRESS.NextPtr = 0x00`

❑ **PCI Express Capabilities**

None

## 2.2.2.2. Default PCI Express Legacy Endpoint

❑ **Standard Upstream Port Type 0 Configuration Header**

❑ **BARs**

```
Bar1.type = Mem32
Bar1.size = 0
Bar2.type = Mem32
Bar2.size = 0
Bar3.type = Mem32
Bar3.size = 0
Bar4.type = Mem32
Bar4.size = 0
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
```

❑ **PCI Capabilities**

```
[Power Management Capability Structure]
PCI_POWER_MANAGEMENT.present = 1
PCI_POWER_MANAGEMENT.offset = 0x40
PCI_POWER_MANAGEMENT.nextcap = 0x50
[MSI-32 Capability Structure]
MSI_32.present = 1
MSI_32.offset = 0x50
MSI_32.nextcap = 0x60
[PCI Express Capabilities Structure]
Type = Legacy EndPoint (Standard Upstream Port PCI Express
Capability Structure)
PCI_EXPRESS.present = 1
PCI_EXPRESS.offset = 0x60
PCI_EXPRESS.NextPtr = 0x00
```

❑ **PCI Express Capabilities**

None

### 2.2.2.3. Default PCI Express Upstream Switch Port

- ❑ Standard Type 1 Configuration Header

- ❑ BARs

```
Bar1.type = Mem64
Bar1.size = 0
```

- ❑ PCI Capabilities

```
[Power Management Capability Structure]
Type = Upstream Switch Port (Standard Type 1 Header Device PMC
Structure)
PCI_POWER_MANAGEMENT.present = 1
PCI_POWER_MANAGEMENT.offset = 0x40
PCI_POWER_MANAGEMENT.nextcap = 0x50
[PCI Express Capabilities Structure]
Type = Upstream Switch Port (Standard Upstream Port PCI Express
Capability Structure)
PCI_EXPRESS.present = 1
PCI_EXPRESS.offset = 0x60
PCI_EXPRESS.NextPtr = 0x00
```

- ❑ PCI Express Capabilities

```
None
```

### 2.2.2.4. Default PCI Express Downstream Switch Port

- ❑ Standard Type 1 Configuration Header

- ❑ BARs

```
Bar1.type = Mem64
Bar1.size = 0
```

- ❑ PCI Capabilities

```
[Power Management Capability Structure]
Type = Downstream Switch Port (Standard Type 1 Header Device PMC
Structure)
PCI_POWER_MANAGEMENT.present = 1
PCI_POWER_MANAGEMENT.offset = 0x40
PCI_POWER_MANAGEMENT.nextcap = 0x50
[PCI Express Capabilities Structure]
Type = Downstream Switch Port (Standard Downstream Port PCI
Express Capability Structure)
PCI_EXPRESS.present = 1
PCI_EXPRESS.offset = 0x60
PCI_EXPRESS.NextPtr = 0x00
```

- ❑ PCI Express Capabilities

```
None
```

### 2.2.2.5. Default PCI Express Downstream Switch Port with Slot

- ❑ Standard Type 1 Configuration Header
- ❑ BARs
  - Bar1.type = Mem64
  - Bar1.size = 0
- ❑ PCI Capabilities
  - [Power Management Capability Structure]
  - Type = Downstream Switch Port (Standard Type 1 Header Device PMC Structure)
  - PCI\_POWER\_MANAGEMENT.present = 1
  - PCI\_POWER\_MANAGEMENT.offset = 0x40
  - PCI\_POWER\_MANAGEMENT.nextcap = 0x50
  - [PCI Express Capabilities Structure]
  - Type = Downstream Switch Port (Standard Downstream Port PCI Express Capability Structure)
  - PCI\_POWER\_MANAGEMENT.present = 1
  - PCI\_POWER\_MANAGEMENT.offset = 0x40
  - PCI\_POWER\_MANAGEMENT.nextcap = 0x50
- ❑ PCI Express Capabilities
  - None

### 2.2.2.6. Default PCI Express Downstream Switch Port with Hot Plug Slot

- ❑ Standard Type 1 Configuration Header
- ❑ BARs
  - Bar1.type = Mem64
  - Bar1.size = 0
- ❑ PCI Capabilities
  - [Power Management Capability Structure]
  - Type = Downstream Switch Port (Standard Type 1 Header Device PMC Structure)
  - PCI\_POWER\_MANAGEMENT.present = 1
  - PCI\_POWER\_MANAGEMENT.offset = 0x40
  - PCI\_POWER\_MANAGEMENT.nextcap = 0x50
  - [PCI Express Capabilities Structure]
  - Type = Downstream Switch Port (Standard Downstream Port PCI Express Capability Structure)
  - PCI\_POWER\_MANAGEMENT.present = 1
  - PCI\_POWER\_MANAGEMENT.offset = 0x40
  - PCI\_POWER\_MANAGEMENT.nextcap = 0x50
- ❑ PCI Express Capabilities
  - None



## 2.2.3. Modifications to Default Devices

Each device in the various test cases in this specification is specified individually. The device starts as one of the default cases described in Section 2.2.2. Exceptions to the default case are listed individually. This section describes the notation used in specifying the each device in the test cases. The notation used is designed to make it simple to parse test cases from text files.

### 2.2.3.1. Device Type

Each device described for a test case has a group of settings under a unique label for that device. The first label used is **[Device 0 Function 0]**.

In general labels have the format **[Device X Function Y]**.

Where **X** is an integer between 0 and 255 and **Y** is an integer between 0 and 7

The default device type for a label is specified as follows:

```
DeviceType = DefaultDeviceType
DefaultDeviceType can have one of the following values:
PCIExpressEndPoint
PCIExpressLegacyEndPoint
PCIExpress1.1LegacyEndPoint - Configuration Default registers
defined in Appendix A
PCIExpress2.0LegacyEndPoint - Configuration Default registers
defined in Appendix B
PCIExpress3.0LegacyEndPoint - Configuration Default registers
defined in Appendix C
PCIExpressUpstreamSwitchPort
PCIExpressDownstreamSwitchPort
PCIExpressDownstreamSwitchPortSlot
PCIExpressDownstreamSwitchPortHotPlugSlot
```

(See Section 2.2.3.7 for additional DeviceType values for switches.)

### 2.2.3.2. Base Address Registers

Base Address registers are often configured for individual test cases. For a type 0 header device the BARs are described using the names:

Bar1, Bar2, Bar3, Bar4, Bar5, Bar6.

Bar Type. There are three different types of BARs.

Mem32, Mem64, IO

Bar Size. The size can be specified in bytes, kilobytes, megabytes, or gigabytes.

**Examples:**

128b	128 bytes
50kb	50*1024 bytes
3mb	3*1024*1024 bytes
2gb	2*1mb*1024 bytes

To specify that Bar1 is a 2mb Memory BAR that is capable of 64 bit addressing the following notation would be used:

```
Bar1.type = Mem64
Bar1.size = 2mb
```

The prefetchable bit may be set individually for a memory bar as indicated in the example below:

```
Bar1.type = Mem32
Bar1.prefetchable = 1
```

IO BARs may or may not have the upper 16 bits hard-wired to zero. This setting is controlled as shown in the following example:

```
Bar1.type = IO
Bar1.upper16 = RO
```

**2.2.3.3. Default Register Values**

Different default values can be specified for any register in a test device configuration space. The Vendor ID and Device ID are common examples that are specified for most test cases. Default values can be specified by register name or by location. Examples of specifying a register default value by name are shown as follows:

```
VENDORID.default = 0x1234
DEVICEID.default = 0xFFFF
```

Examples of making the same specifications by location in configuration space are shown as follows:

```
REG.0x0.16.default = 0x1234
REG.0x2.16.default = 0xFFFF
```

The general formal is REG.OFFSET\_IN\_BYTES\_IN\_HEX.SIZE\_IN\_BITS.default = DEFAULT\_VAL MASK\_IN\_HEX

Mask is optional and is in the form 0xXX. It is used to indicate a limited set of bits that are affected. The default is all bits are affected. The two example requests above could also be stated as follows:

```
REG.0x0.16.default = 0x1234 0xFFFF
REG.0x2.16.default = 0xFFFF 0xFFFF
```

REG indicates an offset is applied to the beginning of configuration space. Offsets can also be specified from the beginning of capabilities by using the keywords listed below for each capability:

- ❑ PCI\_POWER\_MANAGEMENT                      PMC
- ❑ MSI\_32    MSI\_32
- ❑ MSI\_64    MSI\_64
- ❑ MSI\_X
- ❑ PCI\_EXPRESS                                      PCI\_EXP
- ❑ ADVANCED\_ERROR\_REPORTING ADV\_ERR
- ❑ VIRTUAL\_CHANNEL                                VIR\_CHA
- ❑ POWER\_BUDGETING                              POW

A list of the register identifiers (such as VENDORID and DEVICEID) that are commonly used in this specification are given in the table below.

Abbreviated Register Name	General Notation*
VENDORID	REG.0x0.16
DEVICEID	REG.0x2.16
ACCEPTABLE_LOS_LATENCY	REG.0x64.32 0x000001C0
LOS_EXIT_LATENCY	REG.0x6C.32 0x00007000
ACCEPTABLE_L1_LATENCY	REG.0x64.32 0x00000E00
L1_EXIT_LATENCY	REG.0x6C.32 0x00038000

\* Locations are for the default device cases.

### 2.2.3.4. Register Characteristics

Registers have different characteristics in that they can be read only or read write, etc. The different characteristic options are indicated as follows:

- ☐ RO      Read Only
- ☐ RW      Read Write
- ☐ ROS     Read Only Sticky
- ☐ RWS     Read Write Sticky
- ☐ HWINIT Hardware Initialized (Write Once)
- ☐ RW1C    Read Write 1 Clear
- ☐ RW1CS   Read Write 1 Clear Sticky

As an example, the syntax for changing the Vendor ID field to a read-write field is shown in two different formats:

```
VENDORID.characteristic = RW
REG.0x0.8.characteristic = RW
```

### 2.2.3.5. Capabilities (Adding or Removing)

Capabilities may be added or removed from the set that are included in default device cases. The characteristics of the default capability may also need to be modified for some test cases. The different PCI capabilities that can be added or removed are listed as follows:

```
PCI_POWER_MANAGEMENT
MSI_32
MSI_64
MSI_X
PCI_EXPRESS
```

PCI Express capabilities that can be added to default cases are listed as follows:

```
ADVANCED_ERROR_REPORTING
VIRTUAL_CHANNEL
POWER_BUDGETING
```

To add a default MSI\_32 capability the syntax **MSI\_32.present = 1** is used:

To remove a default MSI\_32 capability the **MSI\_32.present = 0** syntax is used:

The offset for a capability and the value of its next capability pointer are frequently specified. The format for these specifications is as show:

```
MSI_32.offset = 0x70
MSI_32.nextcap = 0x80
```

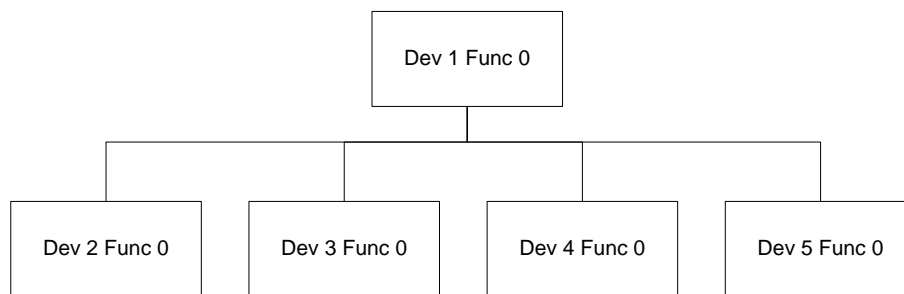
### 2.2.3.6. Bridges and Downstream Devices

If a device is a bridge with downstream devices it must contain an entry specifying the devices on its downstream bus. For example, if Device 0 is a bridge and Device 1 and Device 2 are the downstream bus of the bridge it would be specified as an entry as follows in the section for the bridge device:

```
Bridge = 1,2
```

### 2.2.3.7. Switches

A switch is a group of bridge devices. There is one upstream port and some number of downstream ports. A 4 port switch is shown in the picture below.



The description for this switch would be as follows using the notation described in Section 2.2.2

```

[Device 1 Function 0]
DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = X
DEVICEID = Y
PORTNUMBER = 1
Bridge = 2,3,4,5

[Device 2 Function 0]
DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = X
DEVICEID = Y
PORTNUMBER = 2

[Device 3 Function 0]
DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = X
DEVICEID = Y
PORTNUMBER = 3
  
```

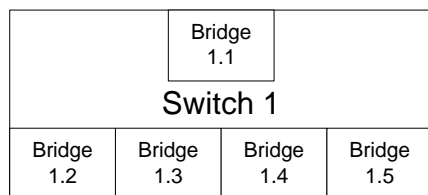
**[Device 4 Function 0]**

```
DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = X
DEVICEID = Y
PORTNUMBER = 4
```

**[Device 5 Function 0]**

```
DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = X
DEVICEID = Y
PORTNUMBER = 5
```

For simplicity the same standard switch can instead be shown and described as follows:

**[Device 1 Function 0]**

```
DeviceType = PCIExpressSwitch
VENDORID = X
DEVICEID = Y
NumberPorts = 4
```

Other options for the switch DeviceType are:

- ☐ PCIExpressSwitch
- ☐ PCIExpressSwitchSlots
- ☐ PCIExpressSwitchHotPlugSlots

This shorthand notation for describing a switch requires the ports to be uniform. The long notation must be used to describe a switch that has differences in registers from one port to the next.

To indicate devices that are downstream of each of the switch downstream ports the bridge syntax is used with a port number appended. For example – to indicate that device 8 is downstream of downstream switch port 1.2 the following entry would be included:

```
Bridge.2 = 8
```

### 2.2.3.8. Expansion Rom

The Expansion ROM base address register is sized in a similar fashion to other base address registers. An example setting the size to 256b is shown below:

```
OptionRomBar.size = 256b
```

Characteristics for the Expansion ROM image are also specified in a number of test cases. The exact images are not specified in this specification. The images have the correct CRC and perform some operation that is visible to the test software. Image initialization and run time size are specified as follows:

```
OptionRomImage.initsize = 20kb
OptionRomImage.runtimesize = 10kb
```

### 2.2.3.9. Command Register Enable Bits

In some test cases devices intentionally make unreasonable resource requests. For example, a device might request several gigabytes of memory in a 32 bit address space. In such cases the platform under test must not enable memory access for this device. To indicate the required behavior for the Command register enable bits for a specific test device, the following notation is used:

```
Bits:
IOEnable
MemoryEnable
BusMasterEnable
Values:
    0 Must Not Be Set
    1 Must Be Set
    2 No Requirement
```

For example, specifying that the Memory Space Enable bit must be set would be done as

```
MemoryEnable = 1.
```

If these settings are not specified for a test device the following defaults apply.

```
If IO is requested:
IOEnable = 1
No IO Requested:
IOEnable = 2
If memory is requested:
MemoryEnable = 1
```

```
If no memory is requested:
MemoryEnable = 2
```

For all cases:

```
BusMasterEnable = 2
```

### 2.2.3.10. Identical Devices

In some cases there are more than one identical device in a test topology. A method for indicating that a device is identical to a previously described device is shown in the example below where Device 2 Function 0 is identical to Device 1 Function 0:

```
[Device 2 Function 0]
DeviceType = Device 1 Function 0
```

### 2.2.3.11. Device Response Characteristics

There are several settings that control how a device responds to configuration requests.

AllBusDevNumResponse controls whether a device will respond to any type 0 configuration cycles regardless of the bus number and device number in the request.

The default is yes (1).

NoResponse make a device unresponsive to all cycles if set. The default is not set (0).

FirstResponseDelay controls how long the device takes to respond to the first configuration cycle after a reset. The time is in milliseconds. The default is as quickly as possible.

ConfigRetry controls settings on the device producing configuration retry responses. Options are listed below.

```
ConfigRetry = First // Only the first configuration request
                    receives a retry
```

```
ConfigRetry = All   // All requests receive a retry.
```

```
ConfigRetry = None  // Default - no configuration retries are
                    used.
```



### 2.2.3.12. Active State Power Management

In some test cases the focus is on monitoring platform configuration of active state power management in various situations. The active state power management acceptable and exit latency register values are configured for these test cases as follows:

```
ACCEPTABLE_L0S_LATENCY.default = X
L0S_EXIT_LATENCY.default = X
```

Where **X** is one of the following values

X	ACCEPTABLE_L0S_LATENCY	L0S_EXIT_LATENCY
0	Maximum of 64 ns	Less than 64 ns
1	Maximum of 128 ns	64 ns to less than 128 ns
2	Maximum of 256 ns	128 ns to less than 256 ns
3	Maximum of 512 ns	256 ns to less than 512 ns
4	Maximum of 1 us	512 ns to less than 1 us
5	Maximum of 2 us	1 us to less than 2 us
6	Maximum of 4 us	2 us to 4 us
7	No Limit	More than 4 us

```
ACCEPTABLE_L1_LATENCY.default = X
L1_EXIT_LATENCY.default = X
```

Where **X** is one of the following values

X	ACCEPTABLE_L1_LATENCY	L1_EXIT_LATENCY
0	Maximum of 1 us	Less than 1 us
1	Maximum of 2 us	1 us to less than 2 us
2	Maximum of 4 us	2 us to less than 4 us
3	Maximum of 8 us	4 us to less than 8 us
4	Maximum of 16 us	8 us to less than 16 us
5	Maximum of 32 us	16 us to less than 32 us
6	Maximum of 64 us	32 us to 64 us
7	No Limit	More than 64 us

The required behavior of the system with regard to enabling active state power management can be specified for individual devices. The following notation is used:

```
ASPML0sEnable = X
```

```
ASPML1Enable = Y
```

Values:

```
0  Must Not Be Enabled
```

```
1  Must Be Enabled
```

```
2  No Requirement
```

The default setting for both ASPML0sEnable and ASPML1Enable is No Requirement.

### 2.2.3.13. Vendor ID and Device ID

Intel has reserved the following Vendor ID and Device ID pair for use by test devices. Any reserved pair of Vendor ID and Device ID could be used.

```
Vendor ID      0x8086
```

```
Device ID      0x6001
```

## **3. Test Cases**

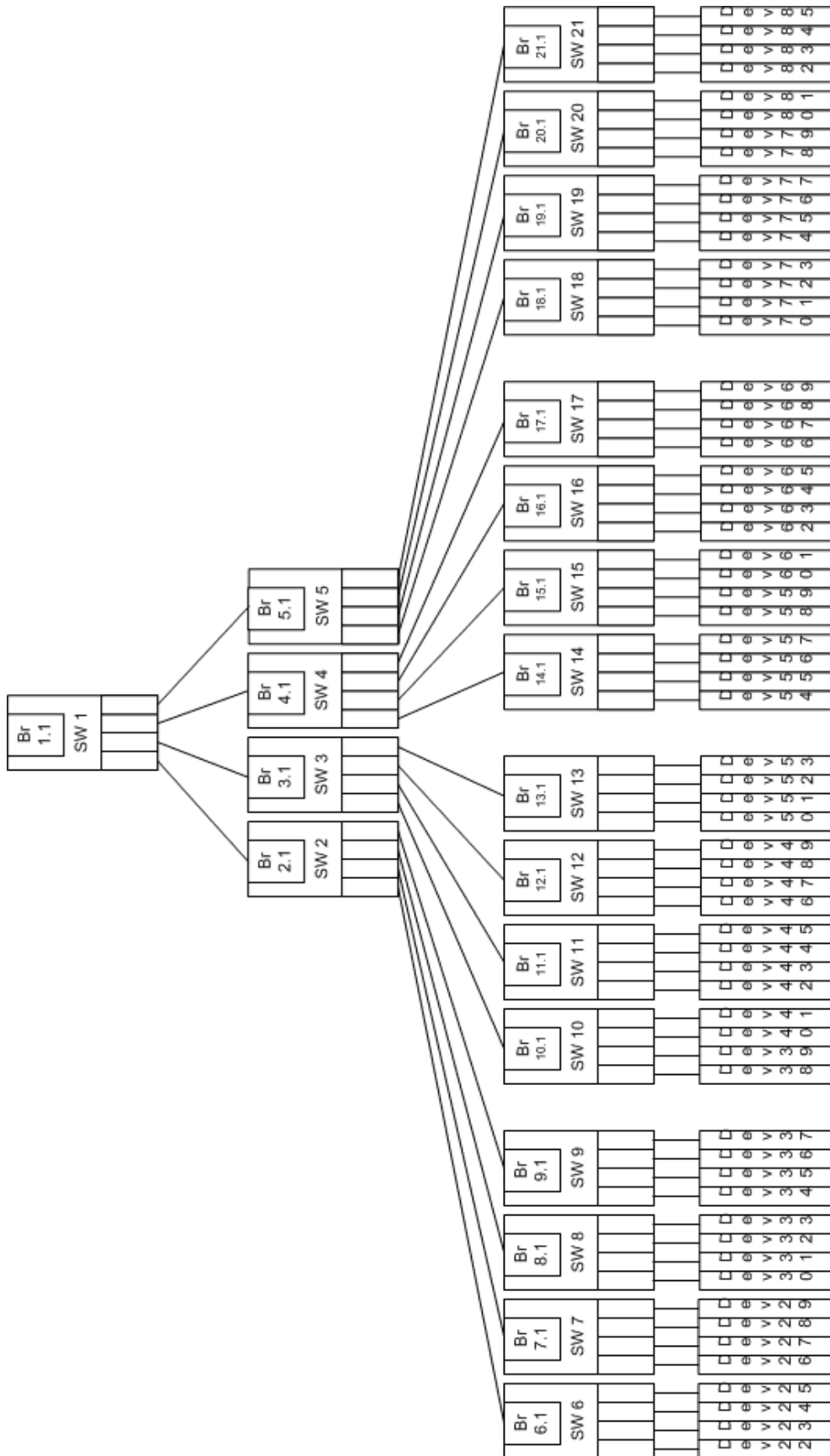
### **3.1. Tests 1.x - Worst Case Complex Topology Support Test Cases**

These test cases use a platform test device(s) to provide PCI Express topologies involving a large number of PCI-PCI bridges. The test makes sure that all devices are found and configured by the platform under test – regardless of location in a complicated topology.

#### **3.1.1. Test 1.1 - Null Case - Informational Only Test**

##### **3.1.1.1. Starting Configuration**

The following test device configuration is used:



**[Device 1 Function 0]**

DeviceType = PCIExpressSwitch  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 2  
Bridge.3 = 3  
Bridge.4 = 4  
Bridge.5 = 5

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = **0x6001**  
NumberPorts = 4  
Bridge.2 = 6  
Bridge.3 = 7  
Bridge.4 = 8  
Bridge.5 = 9

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = **0x6001**  
NumberPorts = 4  
Bridge.2 = 10  
Bridge.3 = 11  
Bridge.4 = 12  
Bridge.5 = 13

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = **0x6001**  
NumberPorts = 4  
Bridge.2 = 14  
Bridge.3 = 15  
Bridge.4 = 16  
Bridge.5 = 17

**[Device 5 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 18  
Bridge.3 = 19  
Bridge.4 = 20  
Bridge.5 = 21

**[Device 6 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 22  
Bridge.3 = 23  
Bridge.4 = 24  
Bridge.5 = 25

**[Device 7 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 26  
Bridge.3 = 27  
Bridge.4 = 28  
Bridge.5 = 29

**[Device 8 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 30  
Bridge.3 = 31  
Bridge.4 = 32  
Bridge.5 = 33

**[Device 9 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 34  
Bridge.3 = 35  
Bridge.4 = 36  
Bridge.5 = 37

**[Device 10 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 38  
Bridge.3 = 39  
Bridge.4 = 40  
Bridge.5 = 41

**[Device 11 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 42  
Bridge.3 = 43  
Bridge.4 = 44  
Bridge.5 = 45

**[Device 12 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 46  
Bridge.3 = 47  
Bridge.4 = 48  
Bridge.5 = 49

**[Device 13 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 50  
Bridge.3 = 51  
Bridge.4 = 52  
Bridge.5 = 53

**[Device 14 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 54  
Bridge.3 = 55  
Bridge.4 = 56  
Bridge.5 = 57

**[Device 15 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 58  
Bridge.3 = 59  
Bridge.4 = 60  
Bridge.5 = 61

**[Device 16 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 62  
Bridge.3 = 63  
Bridge.4 = 64  
Bridge.5 = 65



**[Device 17 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 66  
Bridge.3 = 67  
Bridge.4 = 68  
Bridge.5 = 69

**[Device 18 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 70  
Bridge.3 = 71  
Bridge.4 = 72  
Bridge.5 = 73

**[Device 19 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 74  
Bridge.3 = 75  
Bridge.5 = 76  
Bridge.5 = 77

**[Device 20 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 78  
Bridge.3 = 79  
Bridge.4 = 80  
Bridge.5 = 81

**[Device 21 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 82  
 Bridge.3 = 83  
 Bridge.4 = 84  
 Bridge.5 = 85

**[Device 22 Function 0]**

DeviceType = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 0b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0  
 Bar6.type = Mem32  
 Bar6.size = 0

**[Device 23 Function 0]**

DeviceType = Device 22 Function 0

**[Device 24 Function 0]**

DeviceType = Device 22 Function 0

**[Device 25 Function 0]**

DeviceType = Device 22 Function 0

**[Device 26 Function 0]**

DeviceType = Device 22 Function 0

**[Device 27 Function 0]**

DeviceType = Device 22 Function 0

**[Device 28 Function 0]**

DeviceType = Device 22 Function 0

**[Device 29 Function 0]**

DeviceType = Device 22 Function 0

**[Device 30 Function 0]**

DeviceType = Device 22 Function 0

**[Device 31 Function 0]**

DeviceType = Device 22 Function 0

```
[Device 32 Function 0]
DeviceType = Device 22 Function 0

[Device 33 Function 0]
DeviceType = Device 22 Function 0

[Device 34 Function 0]
DeviceType = Device 22 Function 0

[Device 35 Function 0]
DeviceType = Device 22 Function 0

[Device 36 Function 0]
DeviceType = Device 22 Function 0

[Device 37 Function 0]
DeviceType = Device 22 Function 0

[Device 38 Function 0]
DeviceType = Device 22 Function 0

[Device 39 Function 0]
DeviceType = Device 22 Function 0

[Device 40 Function 0]
DeviceType = Device 22 Function 0

[Device 41 Function 0]
DeviceType = Device 22 Function 0

[Device 42 Function 0]
DeviceType = Device 22 Function 0

[Device 43 Function 0]
DeviceType = Device 22 Function 0

[Device 44 Function 0]
DeviceType = Device 22 Function 0

[Device 45 Function 0]
DeviceType = Device 22 Function 0

[Device 46 Function 0]
DeviceType = Device 22 Function 0

[Device 47 Function 0]
DeviceType = Device 22 Function 0

[Device 48 Function 0]
DeviceType = Device 22 Function 0

[Device 49 Function 0]
DeviceType = Device 22 Function 0

[Device 50 Function 0]
DeviceType = Device 22 Function 0

[Device 51 Function 0]
DeviceType = Device 22 Function 0

[Device 52 Function 0]
DeviceType = Device 22 Function 0
```

```
[Device 53 Function 0]
DeviceType = Device 22 Function 0

[Device 54 Function 0]
DeviceType = Device 22 Function 0

[Device 55 Function 0]
DeviceType = Device 22 Function 0

[Device 56 Function 0]
DeviceType = Device 22 Function 0

[Device 57 Function 0]
DeviceType = Device 22 Function 0

[Device 58 Function 0]
DeviceType = Device 22 Function 0

[Device 59 Function 0]
DeviceType = Device 22 Function 0

[Device 60 Function 0]
DeviceType = Device 22 Function 0

[Device 61 Function 0]
DeviceType = Device 22 Function 0

[Device 62 Function 0]
DeviceType = Device 22 Function 0

[Device 63 Function 0]
DeviceType = Device 22 Function 0

[Device 64 Function 0]
DeviceType = Device 22 Function 0

[Device 65 Function 0]
DeviceType = Device 22 Function 0

[Device 66 Function 0]
DeviceType = Device 22 Function 0

[Device 67 Function 0]
DeviceType = Device 22 Function 0

[Device 68 Function 0]
DeviceType = Device 22 Function 0

[Device 69 Function 0]
DeviceType = Device 22 Function 0

[Device 70 Function 0]
DeviceType = Device 22 Function 0

[Device 71 Function 0]
DeviceType = Device 22 Function 0

[Device 72 Function 0]
DeviceType = Device 22 Function 0

[Device 73 Function 0]
DeviceType = Device 22 Function 0
```

```

[Device 74 Function 0]
DeviceType = Device 22 Function 0

[Device 75 Function 0]
DeviceType = Device 22 Function 0

[Device 76 Function 0]
DeviceType = Device 22 Function 0

[Device 77 Function 0]
DeviceType = Device 22 Function 0

[Device 78 Function 0]
DeviceType = Device 22 Function 0

[Device 79 Function 0]
DeviceType = Device 22 Function 0

[Device 80 Function 0]
DeviceType = Device 22 Function 0

[Device 81 Function 0]
DeviceType = Device 22 Function 0

[Device 82 Function 0]
DeviceType = Device 22 Function 0

[Device 83 Function 0]
DeviceType = Device 22 Function 0

[Device 84 Function 0]
DeviceType = Device 22 Function 0

[Device 85 Function 0]
DeviceType = Device 22 Function 0

```

### 3.1.1.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.

### 3.1.1.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

### 3.1.2. Test 1.2 – Variety of Endpoint Requests - Informational Only Test

All subsequent 3.1.x test cases have the exact same format as 3.1.1. Only the starting configurations are listed.

#### 3.1.2.1. Starting Configuration

This configuration uses endpoints with a variety of different requests. The endpoints are similar to those uses in the 3.2.x test cases. The topology of test devices is identical to the 3.1.1 case.

##### **[Device 1 Function 0]**

```
DeviceType = PCIExpressSwitch
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
```

##### **[Device 2 Function 0]**

```
DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 6
Bridge.3 = 7
Bridge.4 = 8
Bridge.5 = 9
```

##### **[Device 3 Function 0]**

```
DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 10
Bridge.3 = 11
Bridge.4 = 12
Bridge.5 = 13
```

##### **[Device 4 Function 0]**

```
DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 14
```

Bridge.3 = 15  
Bridge.4 = 16  
Bridge.5 = 17

**[Device 5 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 18  
Bridge.3 = 19  
Bridge.4 = 20  
Bridge.5 = 21

**[Device 6 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 22  
Bridge.3 = 23  
Bridge.4 = 24  
Bridge.5 = 25

**[Device 7 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 26  
Bridge.3 = 27  
Bridge.4 = 28  
Bridge.5 = 29

**[Device 8 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 30  
Bridge.3 = 31  
Bridge.4 = 32  
Bridge.5 = 33

**[Device 9 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 34

Bridge.3 = 35  
 Bridge.4 = 36  
 Bridge.5 = 37

**[Device 10 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 38  
 Bridge.3 = 39  
 Bridge.4 = 40  
 Bridge.5 = 41

**[Device 11 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 42  
 Bridge.3 = 43  
 Bridge.4 = 44  
 Bridge.5 = 45

**[Device 12 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 46  
 Bridge.3 = 47  
 Bridge.4 = 48  
 Bridge.5 = 49

**[Device 13 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 50  
 Bridge.3 = 51  
 Bridge.4 = 52  
 Bridge.5 = 53

□ **[Device 14 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 54



Bridge.3 = 55  
Bridge.4 = 56  
Bridge.5 = 57

**[Device 15 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 58  
Bridge.3 = 59  
Bridge.4 = 60  
Bridge.5 = 61

**[Device 16 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 62  
Bridge.3 = 63  
Bridge.4 = 64  
Bridge.5 = 65

**[Device 17 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 66  
Bridge.3 = 67  
Bridge.4 = 68  
Bridge.5 = 69

**[Device 18 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 70  
Bridge.3 = 71  
Bridge.4 = 72  
Bridge.5 = 73

**[Device 19 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 74  
Bridge.3 = 75  
Bridge.5 = 76  
Bridge.5 = 77

**[Device 20 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 78  
Bridge.3 = 79  
Bridge.4 = 80  
Bridge.5 = 81

**[Device 21 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 4  
Bridge.2 = 82  
Bridge.3 = 83  
Bridge.4 = 84  
Bridge.5 = 85

**[Device 22 Function 0]**

**DeviceType** = PCIExpressEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = Mem32  
Bar1.size = 0b  
Bar2.type = Mem32  
Bar2.size = 0b  
Bar3.type = Mem32  
Bar3.size = 0b  
Bar4.type = Mem32  
Bar4.size = 0b  
Bar5.type = Mem32  
Bar5.size = 0  
Bar6.type = Mem32  
Bar6.size = 0

**[Device 23 Function 0]**

DeviceType = Device 22 Function 0

**[Device 24 Function 0]**

DeviceType = Device 22 Function 0

**[Device 25 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 64kb

Bar2.type = IO

Bar2.size = 16b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 0

Bar6.type = Mem32

Bar6.size = 0

**[Device 26 Function 0]**

DeviceType = Device 22 Function 0

**[Device 27 Function 0]**

DeviceType = Device 22 Function 0

**[Device 28 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 64kb

Bar2.type = IO

Bar2.size = 16b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 64kb

Bar6.type = IO

Bar6.size = 16b

**[Device 29 Function 0]**

DeviceType = Device 22 Function 0

**[Device 30 Function 0]**

DeviceType = Device 22 Function 0

**[Device 31 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 128b  
Bar2.type = IO  
Bar2.size = 16b  
Bar3.type = IO  
Bar3.size = 8b  
Bar4.type = IO  
Bar4.size = 4b  
Bar5.type = IO  
Bar5.size = 64b  
Bar6.type = IO  
Bar6.size = 32b

**[Device 32 Function 0]**

DeviceType = Device 22 Function 0

**[Device 33 Function 0]**

DeviceType = Device 22 Function 0

**[Device 34 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 4b  
Bar2.type = IO  
Bar2.size = 8b  
Bar3.type = IO  
Bar3.size = 16b  
Bar4.type = IO  
Bar4.size = 32b  
Bar5.type = IO  
Bar5.size = 64b  
Bar6.type = IO  
Bar6.size = 128b

**[Device 35 Function 0]**

DeviceType = Device 22 Function 0

**[Device 36 Function 0]**

DeviceType = Device 22 Function 0

**[Device 37 Function 0]**

DeviceType = Device 22 Function 0

**[Device 38 Function 0]**

DeviceType = Device 22 Function 0

**[Device 39 Function 0]**

DeviceType = Device 22 Function 0

**[Device 40 Function 0]**

DeviceType = Device 22 Function 0

**[Device 41 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = IO

Bar1.size = 256b

Bar2.type = IO

Bar2.size = 256b

Bar3.type = IO

Bar3.size = 256b

Bar4.type = IO

Bar4.size = 256b

Bar5.type = IO

Bar5.size = 256b

Bar6.type = IO

Bar6.size = 256b

**[Device 42 Function 0]**

DeviceType = Device 22 Function 0

**[Device 43 Function 0]**

DeviceType = Device 22 Function 0

**[Device 44 Function 0]**

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 256b

Bar2.type = Mem32

Bar2.size = 256b

Bar3.type = Mem32

Bar3.size = 256b

Bar4.type = Mem32

Bar4.size = 256b

Bar5.type = Mem32

Bar5.size = 256b

Bar6.type = Mem32

Bar6.size = 256b

**[Device 45 Function 0]**

DeviceType = Device 22 Function 0

**[Device 46 Function 0]**

DeviceType = Device 22 Function 0

**[Device 47 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 16b  
 Bar2.type = Mem32  
 Bar2.size = 256b  
 Bar3.type = Mem32  
 Bar3.size = 4kb  
 Bar4.type = Mem32  
 Bar4.size = 64kb  
 Bar5.type = Mem32  
 Bar5.size = 1mb  
 Bar6.type = Mem32  
 Bar6.size = 16mb

**[Device 48 Function 0]**

DeviceType = Device 22 Function 0

**[Device 49 Function 0]**

DeviceType = Device 22 Function 0

**[Device 50 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 16mb  
 Bar2.type = Mem32  
 Bar2.size = 1mb  
 Bar3.type = Mem32  
 Bar3.size = 64kb  
 Bar4.type = Mem32  
 Bar4.size = 4kb  
 Bar5.type = Mem32  
 Bar5.size = 256kb  
 Bar6.type = Mem32  
 Bar6.size = 16b

**[Device 51 Function 0]**

DeviceType = Device 22 Function 0

**[Device 52 Function 0]**

DeviceType = Device 22 Function 0

**[Device 53 Function 0]**

DeviceType = Device 22 Function 0

**[Device 54 Function 0]**

DeviceType = Device 22 Function 0

**[Device 55 Function 0]**

DeviceType = Device 22 Function 0

**[Device 56 Function 0]**

DeviceType = Device 22 Function 0

**[Device 57 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16b

Bar1.prefetchable = 1

Bar2.type = Mem32

Bar2.size = 256b

Bar2.prefetchable = 1

Bar3.type = Mem32

Bar3.size = 4kb

Bar3.prefetchable = 1

Bar4.type = Mem32

Bar4.size = 64kb

Bar4.prefetchable = 1

Bar5.type = Mem32

Bar5.size = 1mb

Bar5.prefetchable = 1

Bar6.type = Mem32

Bar6.size = 16mb

Bar6.prefetchable = 1

**[Device 58 Function 0]**

DeviceType = Device 22 Function 0

**[Device 59 Function 0]**

DeviceType = Device 22 Function 0

**[Device 60 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16mb

Bar1.prefetchable = 1

Bar2.type = Mem32

Bar2.size = 1mb

Bar2.prefetchable = 1

Bar3.type = Mem32

Bar3.size = 64kb

Bar3.prefetchable = 1

Bar4.type = Mem32

Bar4.size = 4kb

```

Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 256kb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16b
Bar6.prefetchable = 1

```

**[Device 61 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 62 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 63 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

**[Device 64 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 65 Function 0]**

```
DeviceType = Device 22 Function 0
```



**[Device 66 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 4b  
Bar1.upper16 = RO  
Bar2.type = IO  
Bar2.size = 8b  
Bar2.upper16 = RO  
Bar3.type = IO  
Bar3.size = 16b  
Bar3.upper16 = RO  
Bar4.type = IO  
Bar4.size = 32b  
Bar4.upper16 = RO  
Bar5.type = IO  
Bar5.size = 64b  
Bar5.upper16 = RO  
Bar6.type = IO  
Bar6.size = 128b  
Bar6.upper16 = RO

**[Device 67 Function 0]**

DeviceType = Device 22 Function 0

**[Device 68 Function 0]**

DeviceType = Device 22 Function 0

**[Device 69 Function 0]**

DeviceType = Device 22 Function 0

**[Device 70 Function 0]**

DeviceType = Device 22 Function 0

**[Device 71 Function 0]**

DeviceType = Device 22 Function 0

**[Device 72 Function 0]**

DeviceType = Device 22 Function 0

**[Device 73 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem64  
 Bar1.size = 64mb  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0b  
 Bar6.type = Mem32  
 Bar6.size = 0b

**[Device 74 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 1gb  
 Bar2.type = Mem32  
 Bar2.size = 1gb  
 Bar3.type = Mem32  
 Bar3.size = 1gb  
 Bar4.type = Mem32  
 Bar4.size = 1gb  
 Bar5.type = Mem32  
 Bar5.size = 0b  
 Bar6.type = Mem32  
 Bar6.size = 0mb  
 MemoryEnable = 0

**[Device 75 Function 0]**

DeviceType = Device 22 Function 0

**[Device 76 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem64  
 Bar1.size = 64mb  
 Bar1.prefetchable = 1  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0b

```
Bar6.type = Mem32  
Bar6.size = 0b
```

**[Device 77 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 78 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 79 Function 0]**

```
DeviceType = PCIExpressEndPoint
```

```
VENDORID = 0x8086
```

```
DEVICEID = 0x6001
```

```
Bar1.type = Mem32
```

```
Bar1.size = 1gb
```

```
Bar2.type = Mem32
```

```
Bar2.size = 0b
```

```
Bar3.type = Mem32
```

```
Bar3.size = 1gb
```

```
Bar4.type = Mem32
```

```
Bar4.size = 0b
```

```
Bar5.type = Mem32
```

```
Bar5.size = 0b
```

```
Bar6.type = Mem32
```

```
Bar6.size = 0b
```

```
MemoryEnable = 0
```

**[Device 80 Function 0]**

```
DeviceType = Device 22 Function 0
```

**[Device 81 Function 0]**

```
DeviceType = PCIExpressEndPoint
```

```
VENDORID = 0x8086
```

```
DEVICEID = 0x6001
```

```
Bar1.type = Mem32
```

```
Bar1.size = 0b
```

```
Bar2.type = Mem32
```

```
Bar2.size = 1gb
```

```
Bar3.type = Mem32
```

```
Bar3.size = 0b
```

```
Bar4.type = Mem32
```

```
Bar4.size = 0b
```

```
Bar5.type = Mem32
```

```
Bar5.size = 1gb
```

```
Bar6.type = Mem32
```

```
Bar6.size = 0b
```

```
MemoryEnable = 0
```

**[Device 82 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 256b  
Bar2.type = IO  
Bar2.size = 512b  
Bar3.type = IO  
Bar3.size = 1kb  
Bar4.type = IO  
Bar4.size = 2kb  
Bar5.type = IO  
Bar5.size = 4kb  
Bar6.type = Mem32  
Bar6.size = 0b  
IOEnable = 0

**[Device 83 Function 0]**

DeviceType = Device 22 Function 0

**[Device 84 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 4kb  
Bar2.type = IO  
Bar2.size = 2kb  
Bar3.type = IO  
Bar3.size = 1kb  
Bar4.type = IO  
Bar4.size = 512b  
Bar5.type = IO  
Bar5.size = 256b  
Bar6.type = Mem32  
Bar6.size = 0b  
IOEnable = 0

**[Device 85 Function 0]**

DeviceType = Device 22 Function 0

## 3.2. Tests 2.x - PCI-SIG PCI Functional Bios Test Cases

These tests use a configurable two-function PCI Express test device. They cover the same cases currently covered in the PCI-SIG for PCI BIOS testing with the PCI FoxFire test card. These test scenarios are performed by the PCIPOST and PCIEXP utilities for PCI systems.

### 3.2.1. Test 2.1 - Null Case – Required Test

#### 3.2.1.1. Starting Configuration

The following test device configuration is used:

Device 0 Func 0	Device 0 Func 1
-----------------	-----------------

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

### **3.2.1.2. Overview of Test Steps**

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.

### **3.2.1.3. Results Interpretation**

The test fails if any test described in Section 2.1.2 fails.

## 3.2.2. Test 2.2, Simple Case – Required Test

All subsequent 3.2.x test cases have the exact same format as 3.2.1. Only the starting configurations are listed.

## 3.2.3. Base 1.1 Topology

### 3.2.3.1. Starting Configuration

```
[Device 0 Function 0]
DeviceType = PCIExpress1.1LegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

## 3.2.4. Base 2.0 Topology

### 3.2.4.1. Starting Configuration

```
[Device 0 Function 0]
DeviceType = PCIExpress2.0LegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
```

```
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

## 3.2.5. Base 3.0 Topology

### 3.2.5.1. Starting Configuration

```
[Device 0 Function 0]
DeviceType = PCIExpress3.0LegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
```

```
[Device 0 Function 1]
DeviceType = Device 0 Function 0
```



### 3.2.6. Test 2.3: Simple Case with Gaps – Required Test

```
[Device 0 Function 0]
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
VENDORID = X
DEVICEID = Y
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 64kb
Bar6.type = IO
Bar6.size = 16b

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

### 3.2.7. Test 2.4 – A lot of I/O stuff – Required Test

```
[Device 0 Function 0]
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b
Bar3.type = IO
Bar3.size = 8b
Bar4.type = IO
Bar4.size = 4b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 32b
```

```

[Device 0 Function 1]
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b
Bar2.type = IO
Bar2.size = 8b
Bar3.type = IO
Bar3.size = 16b
Bar4.type = IO
Bar4.size = 32b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 128b

```

### 3.2.8. Test 2.5: More I/O Stuff (Checks a lot of 256B I/O requests – Required Test

```

[Device 0 Function 0]
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 256b
Bar2.type = IO
Bar2.size = 256b
Bar3.type = IO
Bar3.size = 256b
Bar4.type = IO
Bar4.size = 256b
Bar5.type = IO
Bar5.size = 256b
Bar6.type = IO
Bar6.size = 256b

```

```

[Device 0 Function 1]
DeviceType = Device 0 Function 0

```

### 3.2.9. Test 2.6: A Bunch Asking for 256B Memory – Required Test

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 256b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 256b
Bar4.type = Mem32
Bar4.size = 256b
Bar5.type = Mem32
Bar5.size = 256b
Bar6.type = Mem32
Bar6.size = 256b

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

### 3.2.10. Test 2.7 – Memory Requests of Various Sizes – Required Test

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 4kb
Bar4.type = Mem32
Bar4.size = 64kb
Bar5.type = Mem32
Bar5.size = 1mb
Bar6.type = Mem32
Bar6.size = 16mb

[Device 0 Function 1]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
```

```

Bar1.type = Mem32
Bar1.size = 16mb
Bar2.type = Mem32
Bar2.size = 1mb
Bar3.type = Mem32
Bar3.size = 64kb
Bar4.type = Mem32
Bar4.size = 4kb
Bar5.type = Mem32
Bar5.size = 256kb
Bar6.type = Mem32
Bar6.size = 16b

```

### 3.2.11. Test 2.8 – Memory Requests of Various Sizes with Prefetchable Bit Set – Required Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 256b
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 4kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 64kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 1mb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16mb
Bar6.prefetchable = 1

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16mb
Bar1.prefetchable = 1
Bar2.type = Mem32

```

```

Bar2.size = 1mb
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 64kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 4kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 256kb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16b
Bar6.prefetchable = 1

```

### 3.2.12. Test 2.9: I/O Requests with top 16 Bits Tied to Zero – Required Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b

```

```

Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 8b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 16b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 32b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 128b
Bar6.upper16 = RO

```

### 3.2.13. Test 2.10: 64-Bit BARs – Required Test

```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b

```

```

[Device 0 Function 1]
DeviceType = Device 0 Function 0

```

### 3.2.14. Test 2.11: 64-Bit BARs with Prefetchable Bit Set – Required Test

```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1

```

```

Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b

```

```

[Device 0 Function 1]
DeviceType = Device 0 Function 0

```

### 3.2.15. Test 2.12: Four BARs Asking for 1GB – Required Test

```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
MemoryEnable = 0

```

```

[Device 0 Function 1]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 1gb
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 1gb
Bar6.type = Mem32

```

```
Bar6.size = 0b  
MemoryEnable = 0
```

### 3.2.16. Test 2.13: I/O Registers Asking for More Than 256 Bytes – Required Test

#### [Device 0 Function 0]

```
DeviceType = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 256b  
Bar2.type = IO  
Bar2.size = 512b  
Bar3.type = IO  
Bar3.size = 1kb  
Bar4.type = IO  
Bar4.size = 2kb  
Bar5.type = IO  
Bar5.size = 4kb  
Bar6.type = Mem32  
Bar6.size = 0b  
IOEnable = 2
```

#### [Device 0 Function 1]

```
DeviceType = PCIExpressLegacyEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bar1.type = IO  
Bar1.size = 4kb  
Bar2.type = IO  
Bar2.size = 2kb  
Bar3.type = IO  
Bar3.size = 1kb  
Bar4.type = IO  
Bar4.size = 512b  
Bar5.type = IO  
Bar5.size = 256b  
Bar6.type = Mem32  
Bar6.size = 0b  
IOEnable = 2
```



### 3.2.17. Test 2.14: One Function Requests Memory of Various Sizes While the Other Makes No Requests – Required Test

#### [Device 0 Function 0]

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 256b
Bar4.type = Mem32
Bar4.size = 4kb
Bar5.type = Mem32
Bar5.size = 64kb
Bar6.type = Mem32
Bar6.size = 1mb
```

#### [Device 0 Function 1]

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
```

### 3.2.18. Test 2.15: One Function Requests Several GB of Memory While the Other Makes Reasonable Memory Requests – Required Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 1gb
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 1gb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0mb
MemoryEnable = 0

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 16b
Bar4.type = Mem32
Bar4.size = 16b
Bar5.type = Mem32
Bar5.size = 16b
Bar6.type = Mem32
Bar6.size = 16b
MemoryEnable=2

```

### 3.2.19. Test 2.16: Simple Expansion ROM Case – Optional Test

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 0b

[Device 0 Function 1]
DeviceType = Device 0 Function 0
```

### 3.2.20. Test 2.17: Function 0 Should Run; Function 1 Should not – Optional Test

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 9kb
OptionRomImage.runtimesize = 9kb
```

```

[Device 0 Function 1]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 0b

```

### 3.2.21. Test 2.18: Function 1 Should Run; Function 0 Should Not – Optional Test

```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 0kb

```

```

[Device 0 Function 1]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b

```

```

Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 9kb
OptionRomImage.runtimesize = 9kb

```

### 3.2.22. Test 2.19: Both Functions Run – Optional Test

#### **[Device 0 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 9kb
OptionRomImage.runtimesize = 8kb

```

#### **[Device 0 Function 1]**

```

DeviceType = Device 0 Function 0

```

### 3.2.23. Test 2.20: Two 34KB Images – Optional Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 512 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 1mb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb

```

### 3.2.24. Test 2.21: Two 34KB Images, ONE SHRINKS to 5KB RUNTIME – Optional Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 2mb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 4mb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb

```

### 3.2.25. Test 2.22 – Two 34KB Images, Other Shrinks to 5KB Runtime – Optional Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 8mb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```



### 3.2.26. Test 2.23: First Function with Zero Runtime Size – Optional Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 0kb

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb

```

### 3.2.27. Test 2.24: Second Function with Zero Runtime Size – Optional Test

**[Device 0 Function 0]**

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 34kb
```

**[Device 0 Function 1]**

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 0kb
```

### 3.2.28. Test 2.25: Two Different Images – Optional Test

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 16.5kb

```

#### [Device 0 Function 1]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 9kb
OptionRomImage.runtimesize = 3.5kb

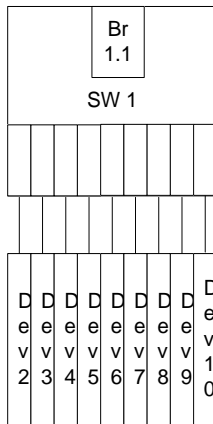
```

### 3.3. Tests 3.x - Expanded Functional Test Cases

These test cases use a platform test device(s) to provide PCI Express topologies involving a large number of PCI-PCI bridges. The test makes sure that all devices are found and configured by the platform under test – regardless of location in a complicated topology.

#### 3.3.1. Test 3.1 - Nine Port Switch with No Requests – OptionalTest

##### 3.3.1.1. Starting Configuration



##### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 9
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
Bridge.6 = 6
Bridge.7 = 7
Bridge.8 = 8
Bridge.9 = 9
Bridge.10 = 10

```

##### [Device 2 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b

```

```

Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

```

**[Device 3 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 4 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 5 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 6 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 7 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 8 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 9 Function 0]**

```
DeviceType = Device 2 Function 0
```

**[Device 10 Function 0]**

```
DeviceType = Device 2 Function 0
```

### 3.3.1.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.

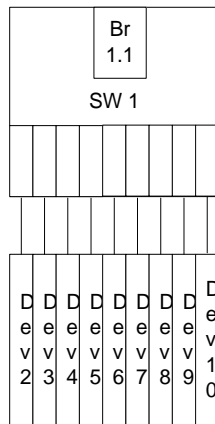
### 3.3.1.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

### 3.3.2. Test 3.2: Nine Port Switch with Various Requests – Optional Test

All subsequent 3.3.x test cases have the exact same format as 3.3.1. Only the starting configurations are listed.

#### 3.3.2.1. Starting Configuration



##### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 9
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
Bridge.6 = 6
Bridge.7 = 7
Bridge.8 = 8
Bridge.9 = 9
Bridge.10 = 10

```

**[Device 2 Function 0]****DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 0b

Bar2.type = Mem32

Bar2.size = 0b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 0

Bar6.type = Mem32

Bar6.size = 0

**[Device 3 Function 0]****DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 64kb

Bar2.type = IO

Bar2.size = 16b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 64kb

Bar6.type = IO

Bar6.size = 16b

**[Device 4 Function 0]****DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = IO

Bar1.size = 128b

Bar2.type = IO

Bar2.size = 16b

Bar3.type = IO

Bar3.size = 8b

Bar4.type = IO

Bar4.size = 4b

Bar5.type = IO

Bar5.size = 64b

Bar6.type = IO

Bar6.size = 32b

**[Device 5 Function 0]**

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16b

Bar2.type = Mem32

Bar2.size = 256b

Bar3.type = Mem32

Bar3.size = 4kb

Bar4.type = Mem32

Bar4.size = 64kb

Bar5.type = Mem32

Bar5.size = 1mb

Bar6.type = Mem32

Bar6.size = 16mb

**[Device 6 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16b

Bar1.prefetchable = 1

Bar2.type = Mem32

Bar2.size = 256b

Bar2.prefetchable = 1

Bar3.type = Mem32

Bar3.size = 4kb

Bar3.prefetchable = 1

Bar4.type = Mem32

Bar4.size = 64kb

Bar4.prefetchable = 1

Bar5.type = Mem32

Bar5.size = 1mb

Bar5.prefetchable = 1

Bar6.type = Mem32

Bar6.size = 16mb

Bar6.prefetchable = 1

**[Device 7 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = IO

Bar1.size = 128b

Bar1.upper16 = RO



```

Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

#### **[Device 8 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 1gb
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 1gb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0mb
MemoryEnable = 0

```

#### **[Device 9 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1
Bar3.type = Mem64
Bar3.size = 16mb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b

```

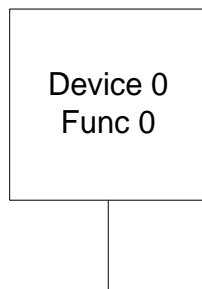
```

[Device 10 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

### 3.3.3. Test 3.3: Bridge with IO Requests - Optional Test

#### 3.3.3.1. Starting Configuration



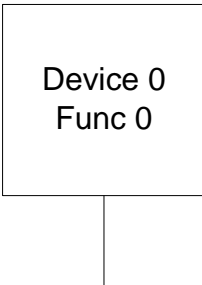
```

[Device 0 Function 0]
DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 0b
Bar2.type = IO
Bar2.size = 256b

```

### 3.3.4. Test 3.4: Bridge with Memory Requests - Optional Test

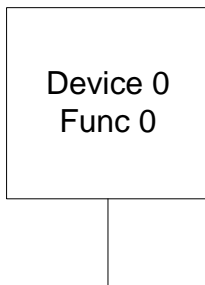
#### 3.3.4.1. Starting Configuration



```
[Device 0 Function 0]
DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 64mb
```

### 3.3.5. Test 3.5: Bridge with 64 Bit Memory Request - Optional Test

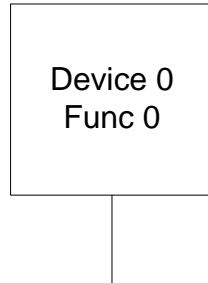
#### 3.3.5.1. Starting Configuration



```
[Device 0 Function 0]
DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
```

### 3.3.6. Test 3.6: Bridge with Expansion ROM - Optional Test

#### 3.3.6.1. Starting Configuration



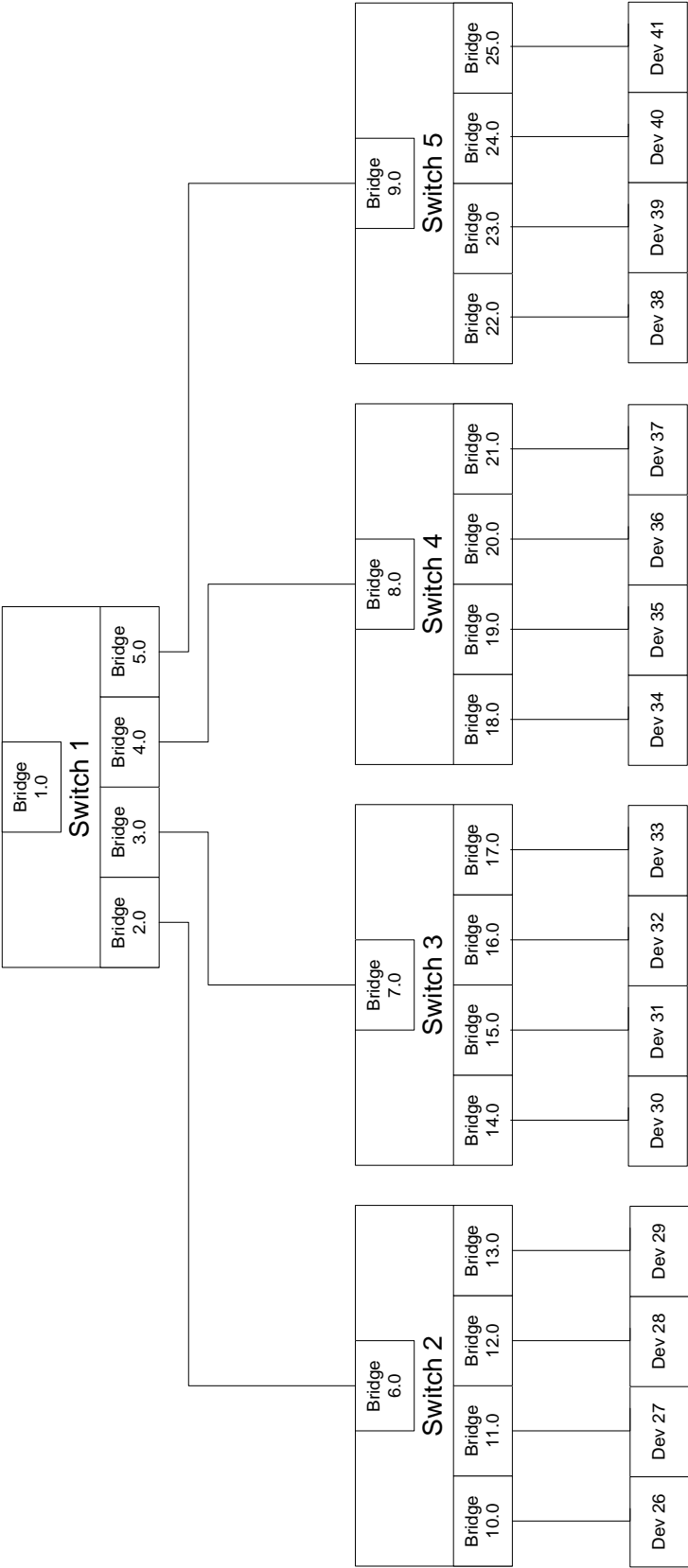
```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
OptionRomBar.size = 2mb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb
  
```

### 3.3.7. Test 3.7 – Multiple Bridges with Various Requests - Informational Only Test

#### 3.3.7.1. Starting Configuration

See drawing on the following page.



**[Device 1 Function 0]**

**DeviceType** = PCIExpressUpstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 1  
Bridge = 2,3,4,5  
Bar1.type = Mem32  
Bar1.size = 0b  
Bar2.type = Mem32  
Bar2.size = 0b

**[Device 2 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 2  
Bridge = 6  
Bar1.type = Mem32  
Bar1.size = 64kb  
Bar2.type = IO  
Bar2.size = 16b

**[Device 3 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 3  
Bridge = 7  
Bar1.type = Mem32  
Bar1.size = 0b  
Bar2.type = Mem32  
Bar2.size = 0b

**[Device 4 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 4  
Bridge = 8  
Bar1.type = Mem32  
Bar1.size = 64kb  
Bar2.type = IO  
Bar2.size = 16b

**[Device 5 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 5

```

Bridge = 9
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b

```

**[Device 6 Function 0]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 10,11,12,13
Bar1.type = IO
Bar1.size = 8b
Bar2.type = IO
Bar2.size = 4b

```

**[Device 7 Function 0]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 14,15,16,17
Bar1.type = IO
Bar1.size = 64b
Bar2.type = IO
Bar2.size = 32b

```

**[Device 8 Function 0]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 18,19,20,21
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b

```

**[Device 9 Function 0]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 22,23,24,25
Bar1.type = Mem32
Bar1.size = 4kb
Bar2.type = Mem32
Bar2.size = 64kb

```

**[Device 10 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 2  
 Bridge = 26  
 Bar1.type = Mem32  
 Bar1.size = 1mb  
 Bar2.type = Mem32  
 Bar2.size = 16mb

**[Device 11 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 3  
 Bridge = 27  
 Bar1.type = Mem32  
 Bar1.size = 16b  
 Bar1.prefetchable = 1  
 Bar2.type = Mem32  
 Bar2.size = 256b  
 Bar2.prefetchable = 1

**[Device 12 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 4  
 Bridge = 28  
 Bar1.type = Mem32  
 Bar1.size = 4kb  
 Bar1.prefetchable = 1  
 Bar2.type = Mem32  
 Bar2.size = 64kb  
 Bar2.prefetchable = 1

**[Device 13 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 5  
 Bridge = 29  
 Bar1.type = Mem32  
 Bar1.size = 1mb  
 Bar1.prefetchable = 1  
 Bar2.type = Mem32  
 Bar2.size = 16mb  
 Bar2.prefetchable = 1



**[Device 14 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 2  
 Bridge = 30  
 Bar1.type = IO  
 Bar1.size = 128b  
 Bar1.upper16 = RO  
 Bar2.type = IO  
 Bar2.size = 16b  
 Bar2.upper16 = RO

**[Device 15 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 3  
 Bridge = 31  
 Bar1.type = IO  
 Bar1.size = 8b  
 Bar1.upper16 = RO  
 Bar2.type = IO  
 Bar2.size = 4b  
 Bar2.upper16 = RO

**[Device 16 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 4  
 Bridge = 32  
 Bar1.type = IO  
 Bar1.size = 64b  
 Bar1.upper16 = RO  
 Bar2.type = IO  
 Bar2.size = 32b  
 Bar2.upper16 = RO

**[Device 17 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 5  
 Bridge = 33  
 Bar1.type = Mem32  
 Bar1.size = 1gb  
 Bar2.type = Mem32  
 Bar2.size = 1gb

MemoryEnable = 0

**[Device 18 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 2  
 Bridge = 34  
 Bar1.type = Mem32  
 Bar1.size = 1gb  
 Bar1.type = Mem32  
 Bar1.size = 1gb  
 MemoryEnable = 0

**[Device 19 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 3  
 Bridge = 35  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 0mb

**[Device 20 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 4  
 Bridge = 36  
 Bar1.type = Mem64  
 Bar1.size = 64mb  
 Bar1.prefetchable = 1

**[Device 21 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 5  
 Bridge = 37  
 Bar1.type = Mem64  
 Bar1.size = 16mb

**[Device 22 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 2  
 Bridge = 38

```

Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b

```

#### **[Device 23 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 3
Bridge = 39
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

#### **[Device 24 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 4
Bridge = 40
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b

```

#### **[Device 25 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 5
Bridge = 41
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b

```

#### **[Device 26 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b

```

```
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 27 Function 0]
DeviceType = Device 26 Function 0

[Device 28 Function 0]
DeviceType = Device 26 Function 0

[Device 29 Function 0]
DeviceType = Device 26 Function 0

[Device 30 Function 0]
DeviceType = Device 26 Function 0

[Device 31 Function 0]
DeviceType = Device 26 Function 0

[Device 32 Function 0]
DeviceType = Device 26 Function 0

[Device 33 Function 0]
DeviceType = Device 26 Function 0

[Device 34 Function 0]
DeviceType = Device 26 Function 0

[Device 35 Function 0]
DeviceType = Device 26 Function 0

[Device 36 Function 0]
DeviceType = Device 26 Function 0

[Device 37 Function 0]
DeviceType = Device 26 Function 0

[Device 38 Function 0]
DeviceType = Device 26 Function 0

[Device 39 Function 0]
DeviceType = Device 26 Function 0

[Device 40 Function 0]
DeviceType = Device 26 Function 0

[Device 41 Function 0]
DeviceType = Device 26 Function 0
```

### 3.3.8. Test 3.8: Eight Function Device with Multiple Requests - Informational Only Test

#### 3.3.8.1. Starting Configuration

Device 0 Func 0	Device 0 Func 1	Device 0 Func 2	Device 0 Func 3	Device 0 Func 4	Device 0 Func 5	Device 0 Func 6	Device 0 Func 7
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

##### [Device 0 Function 0]

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 0b

Bar2.type = Mem32

Bar2.size = 0b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 0

Bar6.type = Mem32

Bar6.size = 0

##### [Device 0 Function 1]

**DeviceType** = PCIExpressLegacyEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 64kb

Bar2.type = IO

Bar2.size = 16b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

Bar5.size = 64kb

Bar6.type = IO

Bar6.size = 16b

**[Device 0 Function 2]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b
Bar3.type = IO
Bar3.size = 8b
Bar4.type = IO
Bar4.size = 4b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 32b

```

**[Device 0 Function 3]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 4kb
Bar4.type = Mem32
Bar4.size = 64kb
Bar5.type = Mem32
Bar5.size = 1mb
Bar6.type = Mem32
Bar6.size = 16mb

```

**[Device 0 Function 4]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 256b
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 4kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 64kb

```

```

Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 1mb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16mb
Bar6.prefetchable = 1

```

#### **[Device 0 Function 5]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

#### **[Device 0 Function 6]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 1gb
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 1gb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0mb
MemoryEnable = 0

```

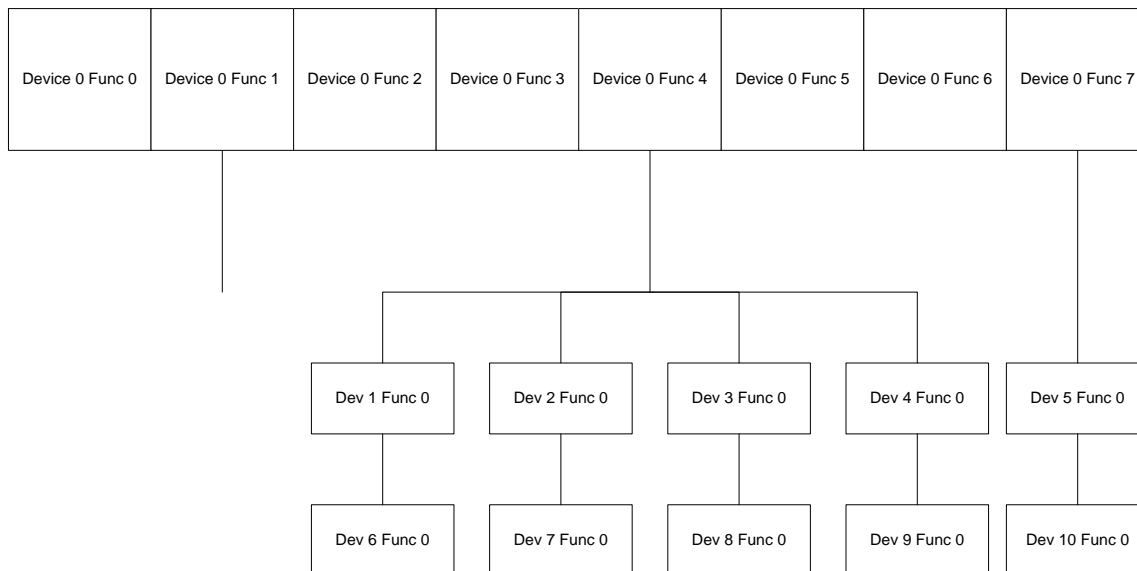
```

[Device 0 Function 7]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1
Bar3.type = Mem64
Bar3.size = 16mb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

### 3.3.9. Test 3.9: Eight Function Device with Type 0 and Type 1 Headers No Resource Requests - Informational Only Test

#### 3.3.9.1. Starting Configuration



```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32

```



```

Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

```

**[Device 0 Function 1]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001

```

**[Device 0 Function 2]**

```

DeviceType = Device 0 Function 0

```

**[Device 0 Function 3]**

```

DeviceType = Device 0 Function 0

```

**[Device 0 Function 4]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 1,2,3,4

```

**[Device 0 Function 5]**

```

DeviceType = Device 0 Function 0

```

**[Device 0 Function 6]**

```

DeviceType = Device 0 Function 0

```

**[Device 0 Function 7]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 5

```

**[Device 1 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 2
Bridge = 6

```

**[Device 2 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 3  
Bridge = 7

**[Device 3 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 4  
Bridge = 8

**[Device 4 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 5  
Bridge = 9

**[Device 5 Function 0]**

**DeviceType** = PCIExpressDownstreamSwitchPort  
VENDORID = 0x8086  
DEVICEID = 0x6001  
PORTNUMBER = 2  
Bridge = 10

**[Device 6 Function 0]**

**DeviceType** = Device 0 Function 0

**[Device 7 Function 0]**

**DeviceType** = Device 0 Function 0

**[Device 8 Function 0]**

**DeviceType** = Device 0 Function 0

**[Device 9 Function 0]**

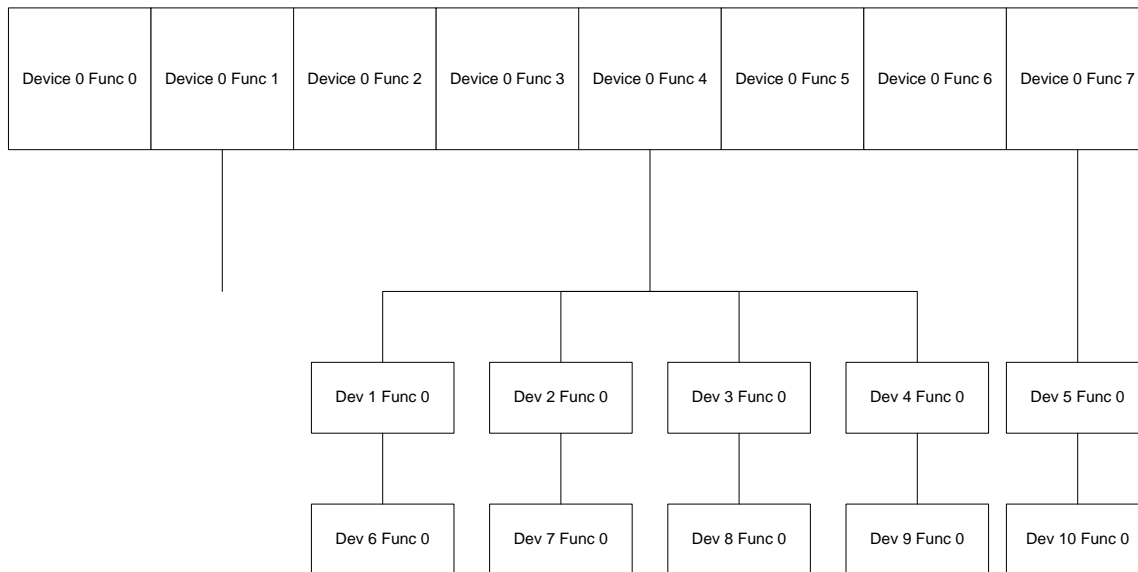
**DeviceType** = Device 0 Function 0

**[Device 10 Function 0]**

**DeviceType** = Device 0 Function 0

### 3.3.10. Test 3.10: Eight Function Device with Type 0 and Type 1 Headers Various Resource Requests - Informational Only Test

#### 3.3.10.1. Starting Configuration



#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
  
```

**[Device 0 Function 1]**

**DeviceType** = PCIExpressUpstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 16mb

**[Device 0 Function 2]**

**DeviceType** = PCIExpressLegacyEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 64kb  
 Bar2.type = IO  
 Bar2.size = 16b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 64kb  
 Bar6.type = IO  
 Bar6.size = 16b

**[Device 0 Function 3]**

**DeviceType** = PCIExpressLegacyEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = IO  
 Bar1.size = 128b  
 Bar2.type = IO  
 Bar2.size = 16b  
 Bar3.type = IO  
 Bar3.size = 8b  
 Bar4.type = IO  
 Bar4.size = 4b  
 Bar5.type = IO  
 Bar5.size = 64b  
 Bar6.type = IO  
 Bar6.size = 32b

**[Device 0 Function 4]**

**DeviceType** = PCIExpressUpstreamSwitchPort  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 PORTNUMBER = 1  
 Bridge = 1,2,3,4

```

Bar1.type = IO
Bar1.size = 0b
Bar2.type = IO
Bar2.size = 256b

```

#### **[Device 0 Function 5]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 4kb
Bar4.type = Mem32
Bar4.size = 64kb
Bar5.type = Mem32
Bar5.size = 1mb
Bar6.type = Mem32
Bar6.size = 16mb

```

#### **[Device 0 Function 6]**

```

DeviceType = PCIExpressUpstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 1
Bridge = 5
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

#### **[Device 1 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 2
Bridge = 6
Bar1.type = Mem32
Bar1.size = 0b
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 0b

```

#### **[Device 2 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 3

```

```

Bridge = 7
Bar1.type = IO
Bar1.size = 0b
Bar2.type = IO
Bar2.size = 256b

```

#### **[Device 3 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 4
Bridge = 8
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 16mb

```

#### **[Device 4 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 5
Bridge = 9
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

#### **[Device 5 Function 0]**

```

DeviceType = PCIExpressDownstreamSwitchPort
VENDORID = 0x8086
DEVICEID = 0x6001
PORTNUMBER = 2
Bridge = 10
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b

```

#### **[Device 6 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO

```

```

Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

#### **[Device 7 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 1gb
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 1gb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0mb
MemoryEnable = 0

```

#### **[Device 8 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1
Bar3.type = Mem64
Bar3.size = 16mb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b

```

#### **[Device 9 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32

```

```
Bar1.size = 0mb
Bar3.type = Mem64
Bar3.size = 16mb
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

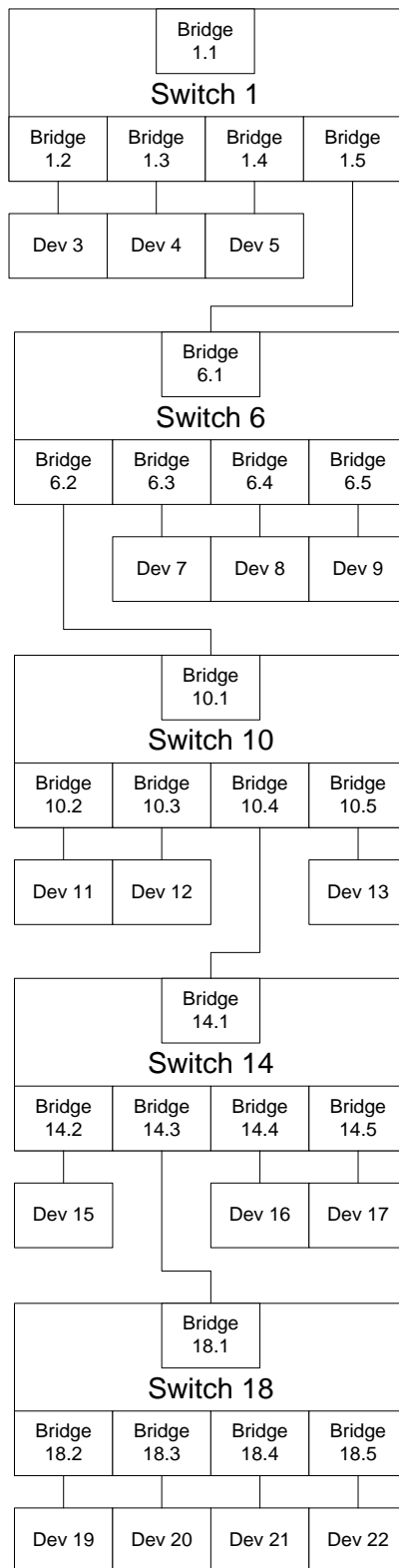
[Device 10 Function 0]
DeviceType = Device 0 Function 0
```

### **3.3.11. Test 3.11: Five Levels of Bridges without Resource Requests - Informational Only Test**

#### **3.3.11.1. Starting Configuration**

See drawing on following page.





**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 3  
 Bridge.3 = 4  
 Bridge.4 = 5  
 Bridge.5 = 6

**[Device 3 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 0b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0  
 Bar6.type = Mem32  
 Bar6.size = 0

**[Device 4 Function 0]**

**DeviceType** = Device 3 Function 0

**[Device 5 Function 0]**

**DeviceType** = Device 3 Function 0

**[Device 6 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 10  
 Bridge.3 = 7  
 Bridge.4 = 8  
 Bridge.5 = 9

**[Device 7 Function 0]**

**DeviceType** = Device 3 Function 0

**[Device 8 Function 0]**

**DeviceType** = Device 3 Function 0

**[Device 9 Function 0]**

**DeviceType** = Device 3 Function 0

**[Device 10 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
NumberPorts = 4  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bridge.2 = 11  
Bridge.3 = 12  
Bridge.4 = 14  
Bridge.5 = 13

**[Device 11 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 12 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 13 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 14 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
NumberPorts = 4  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bridge.2 = 15  
Bridge.3 = 18  
Bridge.4 = 16  
Bridge.5 = 17

**[Device 15 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 16 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 17 Function 0]**  
**DeviceType** = Device 3 Function 0

**[Device 18 Function 0]**  
**DeviceType** = PCIExpressSwitchSlots  
NumberPorts = 4  
VENDORID = 0x8086  
DEVICEID = 0x6001  
Bridge.2 = 19  
Bridge.3 = 20  
Bridge.4 = 21  
Bridge.5 = 22

**[Device 19 Function 0]**  
**DeviceType** = Device 3 Function 0

```
[Device 20 Function 0]
DeviceType = Device 3 Function 0

[Device 21 Function 0]
DeviceType = Device 3 Function 0

[Device 22 Function 0]
DeviceType = Device 3 Function 0
```

### 3.3.12. Test 3.12: Eight Function Device with Only Last Function Implemented - Informational Only Test

Test Note: This is an illegal test case, Function 0 must be implemented and the test should fail if any other function is enabled and Function 0 does not exist.

#### 3.3.12.1. Starting Configuration

Device 0 Func 0	Device 0 Func 1	Device 0 Func 2	Device 0 Func 3	Device 0 Func 4	Device 0 Func 5	Device 0 Func 6	Device 0 Func 7
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
NoResponse = 1

[Device 0 Function 1]
DeviceType = Device 0 Function 0

[Device 0 Function 2]
DeviceType = Device 0 Function 0

[Device 0 Function 3]
DeviceType = Device 0 Function 0

[Device 0 Function 4]
DeviceType = Device 0 Function 0

[Device 0 Function 5]
DeviceType = Device 0 Function 0

[Device 0 Function 6]
DeviceType = Device 0 Function 0
```

```
[Device 0 Function 7]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
```

### 3.3.13. Test 3.13: Eight Function Device with Alternating Functions Implemented - Informational Only Test

Test Note: This is an illegal test case, Function 0 must be implemented and the test should fail if any other function is enabled and Function 0 does not exist.

#### 3.3.13.1. Starting Configuration

Device 0 Func 0	Device 0 Func 1	Device 0 Func 2	Device 0 Func 3	Device 0 Func 4	Device 0 Func 5	Device 0 Func 6	Device 0 Func 7
-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------

```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
NoResponse = 1
```

```
[Device 0 Function 1]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
```

```

Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 0 Function 2]
DeviceType = Device 0 Function 0

[Device 0 Function 3]
DeviceType = Device 0 Function 1

[Device 0 Function 4]
DeviceType = Device 0 Function 0

[Device 0 Function 5]
DeviceType = Device 0 Function 1

[Device 0 Function 6]
DeviceType = Device 0 Function 0

[Device 0 Function 7]
DeviceType = Device 0 Function 1

```

## 3.4. Test 4: Legacy Interrupt Support - Informational Only Test

This test uses a platform test device(s) to generate legacy interrupt messages under the platform port under test. The platform test device is programmed to produce interrupts using all interrupt lines. The timing between interrupt assert and de-assert messages for a pair of interrupts (Assert, De-assert, Assert, De-assert) is varied. The test software verifies that interrupts are received correctly by an ISR for the platform test device in all cases.

### 3.4.1. Starting Configuration

The test device is connected to an available slot on the system under test.

### 3.4.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with a boot device loaded with an operating system that relies on firmware or BIOS support for PCI Express legacy interrupts. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.

4. An interrupt handler is run on the system under test. The interrupt handler writes a designated register on the test device whenever an interrupt is received.
5. The test device generates pairs of Assert INTx and De-assert INTx messages with a variety of timings. De-assert INTx messages are generated whenever the ISR accesses the test device.
6. The test is repeated for each interrupt line.
7. The test is repeated for each available slot on the system under test.
- 0.

### 3.4.3. Results Interpretation

The test fails if the ISR fails to respond to any Assert INTx interrupt message by writing to the test device.

## 3.5. Test 5: Device Ignores PME Fence Mechanism - Informational Only Test

This test is run on downstream ports exposed by the platform under test. A PCI Express Test Device is used on the port under test. The device is programmed to ignore the PME fence mechanism. The test verifies that the system correctly handles a non-responsive device when attempting to remove system power.

### 3.5.1. Starting Configuration

A test device is connected to an available slot on the system under test. The system under test is configured normally.

### 3.5.2. Overview of Test Steps

The test software performs the following steps.

1. Test software prompts the user to connect the platform test device to the slot to be tested.
2. Test software configures the platform test device to not respond to the PME\_Turn\_Off message.
3. The test operator is instructed to shut down the system via operating system methods.
4. The test device records whether the PME\_Turn\_Off message was received.
5. The test device (if supported) or an alternate mechanism is used to verify that system power has been removed.
6. The test is repeated for each available downstream port in the system/switch under test.

### 3.5.3. Results Interpretation

The test fails if the platform test device does not receive the PME\_Turn\_Off message.

The test fails if the platform does not remove power after a device fails to respond to the PME\_Turn\_Off broadcast message.

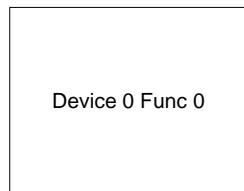
## 3.6. Tests 6.x: Extended Expansion ROM Test Cases

These tests use a configurable PCI Express test device. They cover Expansion ROM configuration test cases for Expansion ROMs that include new optional features in the PCI-SIG Firmware Specification, Revision 3.0.

### 3.6.1. Test 6.1: Device List Pointer - Informational Only Test

#### 3.6.1.1. Starting Configuration

The following test device configuration is used. The test device contains an Expansion ROM that includes a Device List Pointer to a long list of Device IDs for the Expansion ROM. The test checks that the system under test still executes the Expansion ROM correctly.



```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb
```



### 3.6.1.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.
- 0.

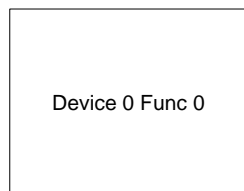
### 3.6.1.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

## 3.6.2. Test 6.2: Maximum Run Time Length - Informational Only Test

### 3.6.2.1. Starting Configuration

The following test device configuration is used. The test device contains an Expansion ROM that contains a Maximum Run Time Length field indicating a length of 5kb.



```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

### 3.6.2.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.

### 3.6.2.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

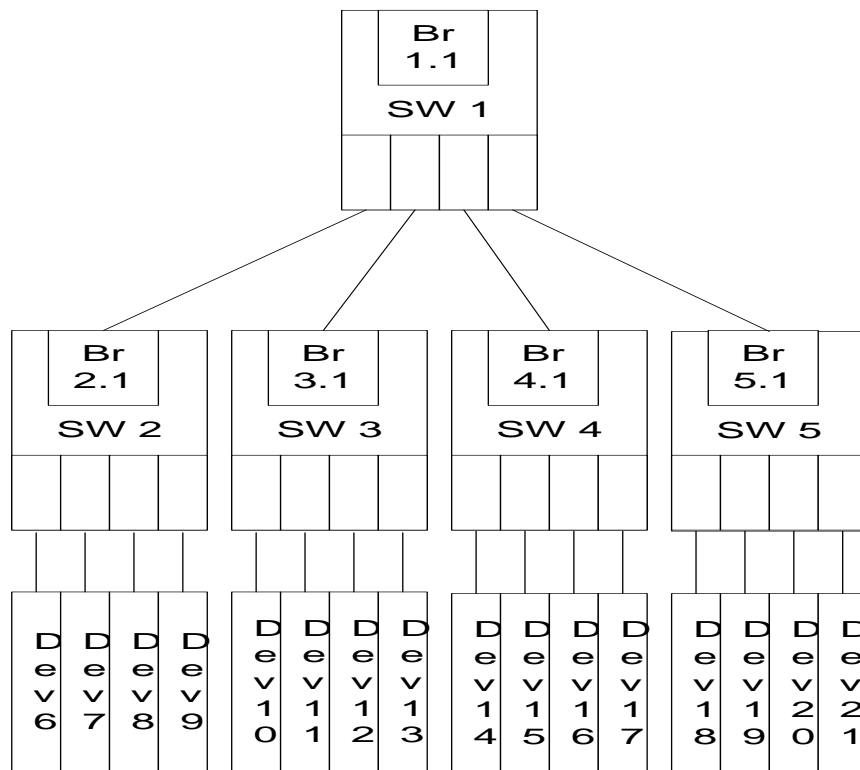
## 3.7. Tests 7.x - Hot Plug Support Test Cases

These test cases use a platform test device(s) to provide PCI Express topologies involving a large number of PCI-PCI bridges. The test makes sure that all devices are found and configured by the platform under test – regardless of location in a complicated topology.

### 3.7.1. Test 7.1: Max Device Null Case - Informational Only Test

#### 3.7.1.1. Starting Configuration

The following test device configuration is used:



#### [Device 1 Function 0]

**DeviceType** = PCIExpressSwitchHotPlugSlots

NumberPorts = 4

VENDORID = 0x8086

DEVICEID = 0x6001

Bridge.1 = 2

Bridge.2 = 3

Bridge.3 = 4  
 Bridge.4 = 5

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchHotPlugSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 6  
 Bridge.3 = 7  
 Bridge.4 = 8  
 Bridge.5 = 9

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchHotPlugSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 10  
 Bridge.3 = 11  
 Bridge.4 = 12  
 Bridge.5 = 13

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchHotPlugSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 14  
 Bridge.3 = 15  
 Bridge.4 = 16  
 Bridge.5 = 17

**[Device 5 Function 0]**

**DeviceType** = PCIExpressSwitchHotPlugSlots  
 NumberPorts = 4  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bridge.2 = 18  
 Bridge.3 = 19  
 Bridge.4 = 20  
 Bridge.5 = 21

**[Device 6 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32

```
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
```

```
[Device 7 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 8 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 9 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 10 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 11 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 12 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 13 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 14 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 15 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 16 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 17 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 18 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 19 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 20 Function 0]
DeviceType = Device 6 Function 0
```

```
[Device 21 Function 0]
DeviceType = Device 6 Function 0
```

### **3.7.1.2. Overview of Test Steps**

Perform the following steps for the test.

1. The system under test is configured with all system devices connected and enabled. The system is powered off. Operating system software for the standard shipping configuration of the system is installed.
2. The system is powered on.
3. The test device(s) is connected to an available hot plug capable slot on the system under test.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.
5. The test is repeated for each hot plug capable slot on the system under test.

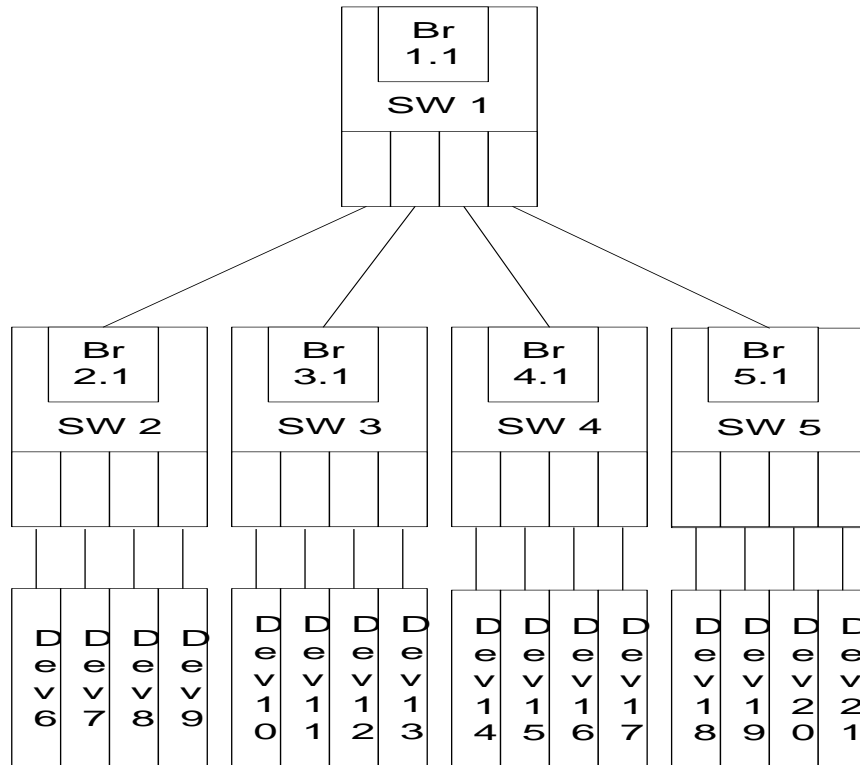
### **3.7.1.3. Results Interpretation**

The test fails if any test described in Section 2.1.2 fails.

### 3.7.2. Test 7.2: Max Devices – Various Requests - Informational Only Test

All subsequent 3.7.x test cases have the exact same format as 3.7.1. Only the starting configurations are listed. This configuration is similar to Test 7.1, except the devices make a variety of memory and IO requests.

#### 3.7.2.1. Starting Configuration



##### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchHotPlugSlots
NumberPorts = 4
VENDORID = 0x8086
DEVICEID = 0x6001
Bridge.1 = 2
Bridge.2 = 3
Bridge.3 = 4
Bridge.4 = 5
  
```

##### [Device 2 Function 0]

```

DeviceType = PCIExpressSwitchHotPlugSlots
NumberPorts = 4
VENDORID = 0x8086
DEVICEID = 0x6001
  
```

```

Bridge.2 = 6
Bridge.3 = 7
Bridge.4 = 8
Bridge.5 = 9

```

**[Device 3 Function 0]**

```

DeviceType = PCIExpressSwitchHotPlugSlots
NumberPorts = 4
VENDORID = 0x8086
DEVICEID = 0x6001
Bridge.2 = 10
Bridge.3 = 11
Bridge.4 = 12
Bridge.5 = 13

```

**[Device 4 Function 0]**

```

DeviceType = PCIExpressSwitchHotPlugSlots
NumberPorts = 4
VENDORID = 0x8086
DEVICEID = 0x6001
Bridge.2 = 14
Bridge.3 = 15
Bridge.4 = 16
Bridge.5 = 17

```

**[Device 5 Function 0]**

```

DeviceType = PCIExpressSwitchHotPlugSlots
NumberPorts = 4
VENDORID = 0x8086
DEVICEID = 0x6001
Bridge.2 = 18
Bridge.3 = 19
Bridge.4 = 20
Bridge.5 = 21

```

**[Device 6 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0

```



```
Bar6.type = Mem32
Bar6.size = 0
```

#### **[Device 7 Function 0]**

```
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
```

#### **[Device 8 Function 0]**

```
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 64kb
Bar6.type = IO
Bar6.size = 16b
```

#### **[Device 9 Function 0]**

```
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b
Bar3.type = IO
Bar3.size = 8b
Bar4.type = IO
Bar4.size = 4b
```

```

Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 32b

```

#### **[Device 10 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b
Bar2.type = IO
Bar2.size = 8b
Bar3.type = IO
Bar3.size = 16b
Bar4.type = IO
Bar4.size = 32b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 128b

```

#### **[Device 11 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 256b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 256b
Bar4.type = Mem32
Bar4.size = 256b
Bar5.type = Mem32
Bar5.size = 256b
Bar6.type = Mem32
Bar6.size = 256b

```

#### **[Device 12 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 256b
Bar2.type = IO
Bar2.size = 256b
Bar3.type = IO
Bar3.size = 256b

```

```

Bar4.type = IO
Bar4.size = 256b
Bar5.type = IO
Bar5.size = 256b
Bar6.type = IO
Bar6.size = 256b

```

**[Device 13 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 4kb
Bar4.type = Mem32
Bar4.size = 64kb
Bar5.type = Mem32
Bar5.size = 1mb
Bar6.type = Mem32
Bar6.size = 16mb

```

**[Device 14 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16mb
Bar2.type = Mem32
Bar2.size = 1mb
Bar3.type = Mem32
Bar3.size = 64kb
Bar4.type = Mem32
Bar4.size = 4kb
Bar5.type = Mem32
Bar5.size = 256kb
Bar6.type = Mem32
Bar6.size = 16b

```

**[Device 15 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar1.prefetchable = 1
Bar2.type = Mem32

```

```

Bar2.size = 256b
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 4kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 64kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 1mb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16mb
Bar6.prefetchable = 1

```

#### **[Device 16 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16mb
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 1mb
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 64kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 4kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 256kb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16b
Bar6.prefetchable = 1

```

#### **[Device 17 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO

```

```

Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

#### **[Device 18 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 8b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 16b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 32b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 128b
Bar6.upper16 = RO

```

#### **[Device 19 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b

```

```
Bar6.type = Mem32
Bar6.size = 0b
MemoryEnable = 0
```

**[Device 20 Function 0]**

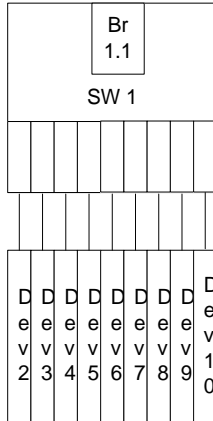
```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
```

**[Device 21 Function 0]**

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
```

### 3.7.3. Test 7.3: Nine Port Switch Null Case - Informational Only Test

#### 3.7.3.1. Starting Configuration



#### [Device 1 Function 0]

**DeviceType** = PCIExpressSwitchHotPlugSlots

NumberPorts = 9

VENDORID = 0x8086

DEVICEID = 0x6001

Bridge.2 = 2

Bridge.3 = 3

Bridge.4 = 4

Bridge.5 = 5

Bridge.6 = 6

Bridge.7 = 7

Bridge.8 = 8

Bridge.9 = 9

Bridge.10 = 10

#### [Device 2 Function 0]

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 0b

Bar2.type = Mem32

Bar2.size = 0b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

```
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 3 Function 0]
DeviceType = Device 2 Function 0

[Device 4 Function 0]
DeviceType = Device 2 Function 0

[Device 5 Function 0]
DeviceType = Device 2 Function 0

[Device 6 Function 0]
DeviceType = Device 2 Function 0

[Device 7 Function 0]
DeviceType = Device 2 Function 0

[Device 8 Function 0]
DeviceType = Device 2 Function 0

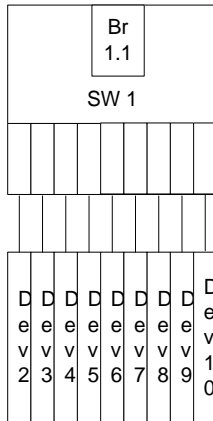
[Device 9 Function 0]
DeviceType = Device 2 Function 0

[Device 10 Function 0]
DeviceType = Device 2 Function 0
```



### 3.7.4. Test 7.4: Nine Port Switch with Multiple Requests- Informational Only Test

#### 3.7.4.1. Starting Configuration



##### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 9
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
Bridge.6 = 6
Bridge.7 = 7
Bridge.8 = 8
Bridge.9 = 9
Bridge.10 = 10

```

##### [Device 2 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 0b
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0

```

```
Bar6.type = Mem32
Bar6.size = 0
```

#### **[Device 3 Function 0]**

```
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 64kb
Bar6.type = IO
Bar6.size = 16b
```

#### **[Device 4 Function 0]**

```
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b
Bar3.type = IO
Bar3.size = 8b
Bar4.type = IO
Bar4.size = 4b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 32b
```

#### **[Device 5 Function 0]**

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 4kb
Bar4.type = Mem32
Bar4.size = 64kb
```

```

Bar5.type = Mem32
Bar5.size = 1mb
Bar6.type = Mem32
Bar6.size = 16mb

```

#### **[Device 6 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 256b
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 4kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 64kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 1mb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16mb
Bar6.prefetchable = 1

```

#### **[Device 7 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b

```

Bar6.upper16 = RO

**[Device 8 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 1gb  
 Bar2.type = Mem32  
 Bar2.size = 1gb  
 Bar3.type = Mem32  
 Bar3.size = 1gb  
 Bar4.type = Mem32  
 Bar4.size = 1gb  
 Bar5.type = Mem32  
 Bar5.size = 0b  
 Bar6.type = Mem32  
 Bar6.size = 0mb  
 MemoryEnable = 0

**[Device 9 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem64  
 Bar1.size = 64mb  
 Bar1.prefetchable = 1  
 Bar3.type = Mem64  
 Bar3.size = 16mb  
 Bar5.type = Mem32  
 Bar5.size = 0b  
 Bar6.type = Mem32  
 Bar6.size = 0b

**[Device 10 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 0b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0b  
 Bar6.type = Mem32

```

Bar6.size = 0b
OptionRomBar.size = 256 kb
OptionRomImage.initsize = 34kb
OptionRomImage.runtimesize = 5kb

```

### 3.7.5. Test 7.5: Simple Five Level Switch Case - Informational Only Test

#### 3.7.5.1. Starting Configuration

See the following drawing.

```

[Device 1 Function 0]
DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 2

```

```

[Device 2 Function 0]
DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 3

```

```

[Device 3 Function 0]
DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 4

```

```

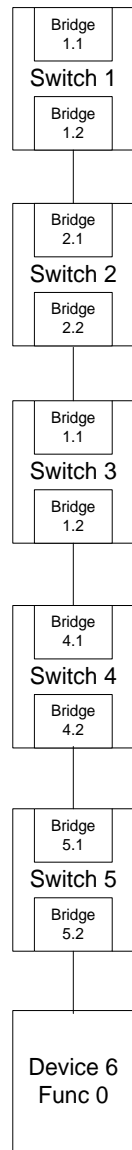
[Device 4 Function 0]
DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 5

```

```

[Device 5 Function 0]
DeviceType = PCIExpressSwitchHotPlugSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 6

```



#### [Device 6 Function 0]

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 0b

Bar2.type = Mem32

Bar2.size = 0b

Bar3.type = Mem32

Bar3.size = 0b

Bar4.type = Mem32

Bar4.size = 0b

Bar5.type = Mem32

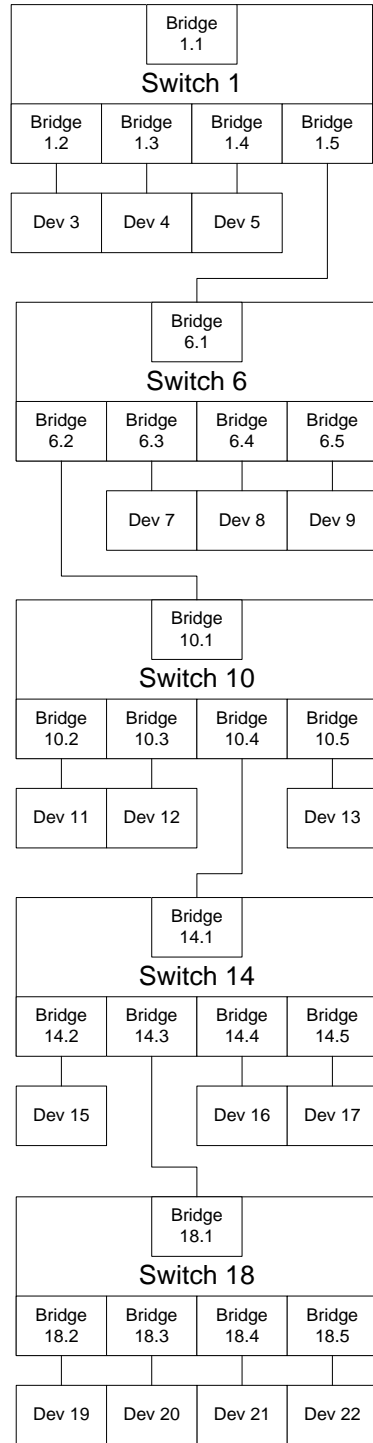
Bar5.size = 0

Bar6.type = Mem32

Bar6.size = 0

### 3.7.6. Test 7.6: Five Levels of Bridges with Various Requests - Informational Only Test

#### 3.7.6.1. Starting Configuration



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 3  
 Bridge.3 = 4  
 Bridge.4 = 5  
 Bridge.5 = 6

**[Device 3 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 0b  
 Bar2.type = Mem32  
 Bar2.size = 0b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0  
 Bar6.type = Mem32  
 Bar6.size = 0

**[Device 4 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32  
 Bar1.size = 64kb  
 Bar2.type = IO  
 Bar2.size = 16b  
 Bar3.type = Mem32  
 Bar3.size = 0b  
 Bar4.type = Mem32  
 Bar4.size = 0b  
 Bar5.type = Mem32  
 Bar5.size = 0  
 Bar6.type = Mem32  
 Bar6.size = 0

**[Device 5 Function 0]**

**DeviceType** = PCIExpressLegacyEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 Bar1.type = Mem32



```

Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 64kb
Bar6.type = IO
Bar6.size = 16b

```

#### **[Device 6 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 10
Bridge.3 = 7
Bridge.4 = 8
Bridge.5 = 9

```

#### **[Device 7 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar2.type = IO
Bar2.size = 16b
Bar3.type = IO
Bar3.size = 8b
Bar4.type = IO
Bar4.size = 4b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 32b

```

#### **[Device 8 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b
Bar2.type = IO
Bar2.size = 8b
Bar3.type = IO
Bar3.size = 16b

```

```

Bar4.type = IO
Bar4.size = 32b
Bar5.type = IO
Bar5.size = 64b
Bar6.type = IO
Bar6.size = 128b

```

#### **[Device 9 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 256b
Bar2.type = Mem32
Bar2.size = 256b
Bar3.type = Mem32
Bar3.size = 256b
Bar4.type = Mem32
Bar4.size = 256b
Bar5.type = Mem32
Bar5.size = 256b
Bar6.type = Mem32
Bar6.size = 256b

```

#### **[Device 10 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 11
Bridge.3 = 12
Bridge.4 = 14
Bridge.5 = 13

```

#### **[Device 11 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 256b
Bar2.type = IO
Bar2.size = 256b
Bar3.type = IO
Bar3.size = 256b
Bar4.type = IO
Bar4.size = 256b
Bar5.type = IO
Bar5.size = 256b
Bar6.type = IO

```

Bar6.size = 256b

**[Device 12 Function 0]**

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16b

Bar2.type = Mem32

Bar2.size = 256b

Bar3.type = Mem32

Bar3.size = 4kb

Bar4.type = Mem32

Bar4.size = 64kb

Bar5.type = Mem32

Bar5.size = 1mb

Bar6.type = Mem32

Bar6.size = 16mb

**[Device 13 Function 0]**

**DeviceType** = PCIExpressEndPoint

VENDORID = 0x8086

DEVICEID = 0x6001

Bar1.type = Mem32

Bar1.size = 16mb

Bar2.type = Mem32

Bar2.size = 1mb

Bar3.type = Mem32

Bar3.size = 64kb

Bar4.type = Mem32

Bar4.size = 4kb

Bar5.type = Mem32

Bar5.size = 256kb

Bar6.type = Mem32

Bar6.size = 16b

**[Device 14 Function 0]**

**DeviceType** = PCIExpressSwitchSlots

VENDORID = 0x8086

DEVICEID = 0x6001

NumberPorts = 4

Bridge.2 = 15

Bridge.3 = 18

Bridge.4 = 16

Bridge.5 = 17

**[Device 15 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16b
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.

```

**[Device 16 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 16mb
Bar1.prefetchable = 1
Bar2.type = Mem32
Bar2.size = 1mb
Bar2.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 64kb
Bar3.prefetchable = 1
Bar4.type = Mem32
Bar4.size = 4kb
Bar4.prefetchable = 1
Bar5.type = Mem32
Bar5.size = 256kb
Bar5.prefetchable = 1
Bar6.type = Mem32
Bar6.size = 16b
Bar6.prefetchable = 1

```

**[Device 17 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 128b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 16b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 8b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 4b
Bar4.upper16 = RO

```

```

Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 32b
Bar6.upper16 = RO

```

**[Device 18 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 19
Bridge.3 = 20
Bridge.4 = 21
Bridge.5 = 22

```

**[Device 19 Function 0]**

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = IO
Bar1.size = 4b
Bar1.upper16 = RO
Bar2.type = IO
Bar2.size = 8b
Bar2.upper16 = RO
Bar3.type = IO
Bar3.size = 16b
Bar3.upper16 = RO
Bar4.type = IO
Bar4.size = 32b
Bar4.upper16 = RO
Bar5.type = IO
Bar5.size = 64b
Bar5.upper16 = RO
Bar6.type = IO
Bar6.size = 128b
Bar6.upper16 = RO

```

**[Device 20 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 1gb
Bar2.type = Mem32
Bar2.size = 0b
Bar3.type = Mem32

```

```
Bar3.size = 1gb
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
MemoryEnable = 0
```

**[Device 21 Function 0]**

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
```

**[Device 22 Function 0]**

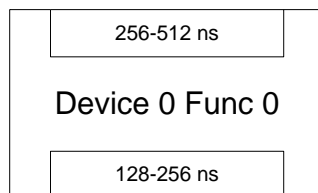
```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem64
Bar1.size = 64mb
Bar1.prefetchable = 1
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0b
Bar6.type = Mem32
Bar6.size = 0b
```

## 3.8. Tests 8.x: L0s Active State PM Enabling In Complex Topologies Test Cases

This test is run on downstream ports exposed by the platform under test. These test cases use a platform test device(s) to provide PCI Express topologies involving a large number of PCI-PCI bridges. The test makes sure that all devices are found and configured by the platform under test – regardless of location in a complicated topology. The device(s) are programmed to represent several different multi-switch topologies with different L0s latencies and L0s latency tolerances. The test verifies that if the platform enables L0s active state PM it does so consistently in all legal cases and does not enable L0s active state PM in incorrect situations.

### 3.8.1. Test 8.1: Single Device with Latency Mismatch - Informational Only Test

#### 3.8.1.1. Starting Configuration



```
[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L0S_LATENCY.default = 2
L0S_EXIT_LATENCY.default = 3
ASPM_L0sEnable = 0
```

#### 3.8.1.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.
5. The test software checks the ASPM configuration to see if it is acceptable for the test case.

### 3.8.1.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

## 3.8.2. Test 8.2: Multi-Function Device with Latency Mismatch in One Function - Informational Only Test

All subsequent 3.8.x test cases have the exact same format as 3.8.1. Only the starting configurations are listed.

### 3.8.2.1. Starting Configuration

<64ns	<64ns	<64ns	<64ns	256-512 ns	<64ns	<64ns	<64ns
Device 0 Func 0	Device 0 Func 1	Device 0 Func 2	Device 0 Func 3	Device 0 Func 4	Device 0 Func 5	Device 0 Func 6	Device 0 Func 7
128-256 ns	128-256 ns	128-256 ns	128-256 ns	128-256 ns	128-256 ns	128-256 ns	128-256 ns

#### [Device 0 Function 0]

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 0
ASPML0sEnable = 2
```

#### [Device 0 Function 1]

```
DeviceType = Device 0 Function 0
```

#### [Device 0 Function 2]

```
DeviceType = Device 0 Function 0
```

#### [Device 0 Function 3]

```
DeviceType = Device 0 Function 0
```

#### [Device 0 Function 4]

```
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 3
ASPML0sEnable = 0
```

#### [Device 0 Function 5]

```
DeviceType = Device 0 Function 0
```

#### [Device 0 Function 6]

```
DeviceType = Device 0 Function 0
```

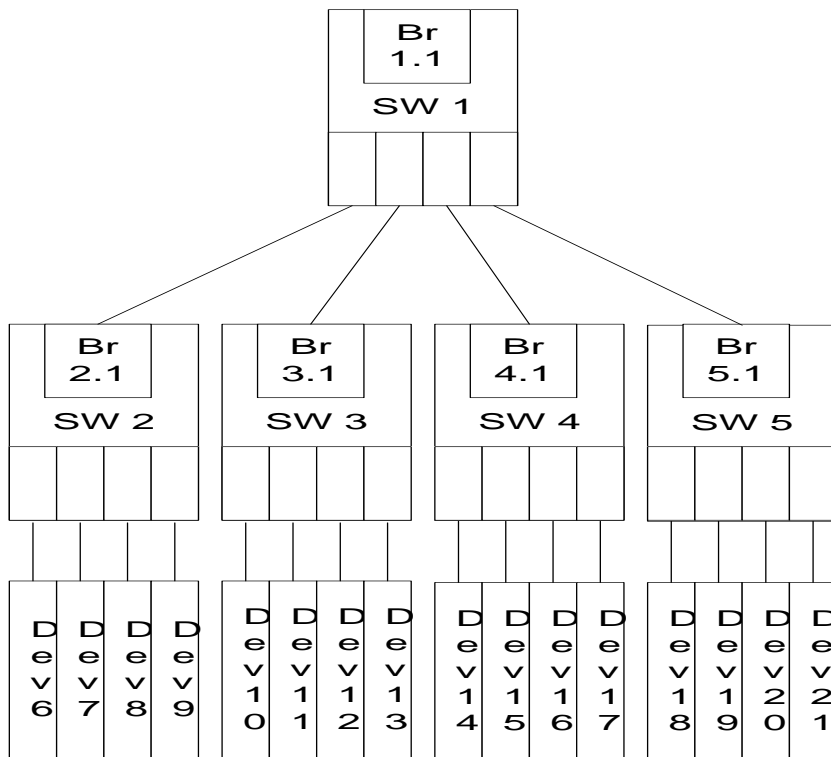
#### [Device 0 Function 7]

```
DeviceType = Device 0 Function 0
```



### 3.8.3. Test 8.3: One Device and One Switch Have Mismatches in Large Topology - Informational Only Test

#### 3.8.3.1. Starting Configuration



#### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 0
ASPML0sEnable = 2

```

**[Device 2 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 6
Bridge.3 = 7
Bridge.4 = 8
Bridge.5 = 9
ACCEPTABLE_L0S_LATENCY.default = 2
L0S_EXIT_LATENCY.default = 0
ASPML0sEnable = 2

```

**[Device 3 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 10
Bridge.3 = 11
Bridge.4 = 12
Bridge.5 = 13
ACCEPTABLE_L0S_LATENCY.default = 2
L0S_EXIT_LATENCY.default = 0
ASPML0sEnable = 2

```

**[Device 4 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 14
Bridge.3 = 15
Bridge.4 = 16
Bridge.5 = 17
ACCEPTABLE_L0S_LATENCY.default = 2
L0S_EXIT_LATENCY.default = 3
ASPML0sEnable = 0

```

```

[Device 5 Function 0]
DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 18
Bridge.3 = 19
Bridge.4 = 20
Bridge.5 = 21
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 0
ASPML0sEnable = 2

```

```

[Device 6 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 0
ASPML0sEnable = 2

```

```

[Device 7 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 8 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 9 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 10 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 11 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 3
ASPML0sEnable = 0

```

```

[Device 12 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 13 Function 0]
DeviceType = Device 6 Function 0

```

```

[Device 14 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_LOS_LATENCY.default = 2
LOS_EXIT_LATENCY.default = 0
ASPML0sEnable = 0

[Device 15 Function 0]
DeviceType = Device 14 Function 0

[Device 16 Function 0]
DeviceType = Device 14 Function 0

[Device 17 Function 0]
DeviceType = Device 14 Function 0

[Device 18 Function 0]
DeviceType = Device 6 Function 0

[Device 19 Function 0]
DeviceType = Device 6 Function 0

[Device 20 Function 0]
DeviceType = Device 6 Function 0

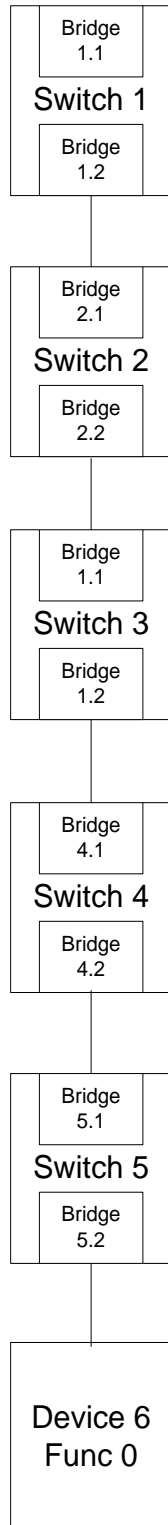
[Device 21 Function 0]
DeviceType = Device 6 Function 0

```

### 3.8.4. Test 8.4: One Switch in Path Creates Latency Mismatch - Informational Only Test

#### 3.8.4.1. Starting Configuration

See drawing on the following page.



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 2  
 ACCEPTABLE\_L0S\_LATENCY.default = 2  
 L0S\_EXIT\_LATENCY.default = 0  
 ASPML0sEnable = 2

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 3  
 ACCEPTABLE\_L0S\_LATENCY.default = 2  
 L0S\_EXIT\_LATENCY.default = 3  
 ASPML0sEnable = 0

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 4  
 ACCEPTABLE\_L0S\_LATENCY.default = 6  
 L0S\_EXIT\_LATENCY.default = 0  
 ASPML0sEnable = 2

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 5  
 ACCEPTABLE\_L0S\_LATENCY.default = 6  
 L0S\_EXIT\_LATENCY.default = 0  
 ASPML0sEnable = 2

**[Device 5 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 6  
 ACCEPTABLE\_L0S\_LATENCY.default = 6  
 L0S\_EXIT\_LATENCY.default = 0  
 ASPML0sEnable = 2

```

[Device 6 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L0S_LATENCY.default = 2
L0S_EXIT_LATENCY.default = 0
ASPML0sEnable = 0

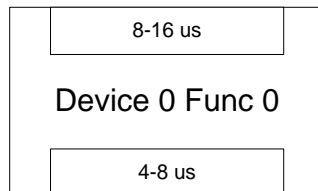
```

## 3.9. Tests 9.x - L1 Active State PM Enabling In Complex Topologies Test Cases

This test is run on downstream ports exposed by the platform under test. These test cases use a platform test device(s) to provide PCI Express topologies involving a large number of PCI-PCI bridges. The test makes sure that all devices are found and configured by the platform under test – regardless of location in a complicated topology. The device(s) are programmed to represent several different multi-switch topologies with different L1 latencies and L1 latency tolerances. The test verifies that if the platform enables L1 active state PM it does so consistently in all legal cases and does not enable L1 active state PM in incorrect situations.

### 3.9.1. Test 9.1: Single Device with Latency Mismatch - Informational Only Test

#### 3.9.1.1. Starting Configuration



```

[Device 0 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 4
ASPML1Enable = 0

```

### 3.9.1.2. Overview of Test Steps

Perform the following steps for the test.

1. The system under test is configured with no boot devices connected. All other system devices are connected and enabled. The system is powered off.
2. The test device(s) is connected to an available slot on the system under test.
3. The system under test is powered on.
4. The test software monitors traffic to the configured topology and performs all checks described in Section 2.1.2.
5. The test software checks the ASPM configuration to see if it is acceptable for the test case.

### 3.9.1.3. Results Interpretation

The test fails if any test described in Section 2.1.2 fails.

## 3.9.2. Test 9.2: Multi-Function Device with Latency Mismatch in One Function - Informational Only Test

All subsequent 3.9.x test cases have the exact same format as 3.9.1. Only the starting configurations are listed.

### 3.9.2.1. Starting Configuration

<1us	<1us	<1us	<1us	8-16 us	<1us	<1us	<1us
Device 0 Func 0	Device 0 Func 1	Device 0 Func 2	Device 0 Func 3	Device 0 Func 4	Device 0 Func 5	Device 0 Func 6	Device 0 Func 7
4-8 us	4-8 us	4-8 us	4-8 us	4-8 us	4-8 us	4-8 us	4-8 us

#### [Device 0 Function 0]

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 0
ASPML1Enable = 2

```

#### [Device 0 Function 1]

```

DeviceType = Device 0 Function 0

```

#### [Device 0 Function 2]

```

DeviceType = Device 0 Function 0

```

#### [Device 0 Function 3]

```

DeviceType = Device 0 Function 0

```



```

[Device 0 Function 4]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 4
ASPML1Enable = 2

```

```

[Device 0 Function 5]
DeviceType = Device 0 Function 0

```

```

[Device 0 Function 6]
DeviceType = Device 0 Function 0

```

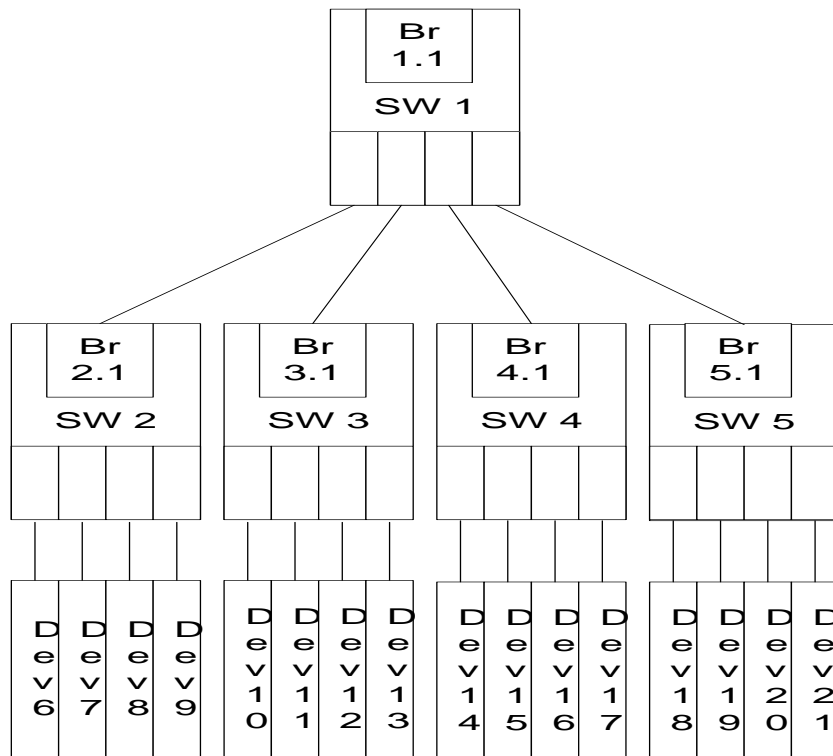
```

[Device 0 Function 7]
DeviceType = Device 0 Function 0

```

### 3.9.3. Test 9.3: One Device and One Switch Have Mismatches in Large Topology - Informational Only Test

#### 3.9.3.1. Starting Configuration



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 2  
 Bridge.3 = 3  
 Bridge.4 = 4  
 Bridge.5 = 5  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 6  
 Bridge.3 = 7  
 Bridge.4 = 8  
 Bridge.5 = 9  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 10  
 Bridge.3 = 11  
 Bridge.4 = 12  
 Bridge.5 = 13  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 4  
 Bridge.2 = 14  
 Bridge.3 = 15  
 Bridge.4 = 16  
 Bridge.5 = 17  
 ACCEPTABLE\_L0S\_LATENCY.default = 3

```

LOS_EXIT_LATENCY.default = 4
ASPML1Enable = 0

```

**[Device 5 Function 0]**

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 18
Bridge.3 = 19
Bridge.4 = 20
Bridge.5 = 21
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 0
ASPML1Enable = 2

```

**[Device 6 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 0
ASPML1Enable = 2

```

**[Device 7 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 8 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 9 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 10 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 11 Function 0]DeviceType = PCIExpressEndPoint**

```

VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 4
ASPML1Enable = 0

```

**[Device 12 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 13 Function 0]**

```

DeviceType = Device 6 Function 0

```

**[Device 14 Function 0]**

```

DeviceType = PCIExpressEndPoint
VENDORID = 0x8086

```

```

DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 0
ASPML1Enable = 0

[Device 15 Function 0]
DeviceType = Device 14 Function 0

[Device 16 Function 0]
DeviceType = Device 14 Function 0

[Device 17 Function 0]
DeviceType = Device 14 Function 0

[Device 18 Function 0]
DeviceType = Device 6 Function 0

[Device 19 Function 0]
DeviceType = Device 6 Function 0

[Device 20 Function 0]
DeviceType = Device 6 Function 0

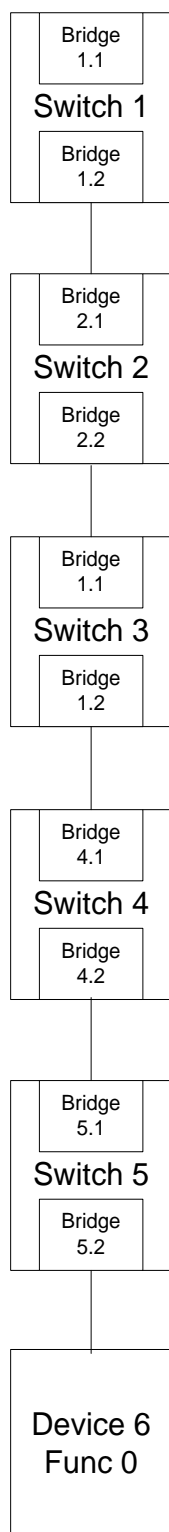
[Device 21 Function 0]
DeviceType = Device 6 Function 0

```

### **3.9.4. Test 9.4 - One Switch in Path Creates Latency Mismatch - Informational Only Test**

#### **3.9.4.1. Starting Configuration**

See drawing on the following page.



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 2  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML0sEnable = 2

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 3  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 4  
 ASPML1Enable = 0

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 4  
 ACCEPTABLE\_L1\_LATENCY.default = 6  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 5  
 ACCEPTABLE\_L1\_LATENCY.default = 6  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

```

[Device 5 Function 0]
DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 1
Bridge.2 = 6
ACCEPTABLE_L1_LATENCY.default = 6
L1_EXIT_LATENCY.default = 0
ASPML1Enable = 2

```

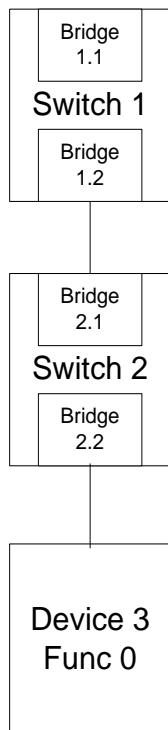
```

[Device 6 Function 0]
DeviceType = PCIExpressEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
ACCEPTABLE_L1_LATENCY.default = 3
L1_EXIT_LATENCY.default = 0
ASPML0sEnable = 0

```

### 3.9.5. Test 9.5: Two Switch Delays Cause Potential L1 Mismatch - Informational Only Test

#### 3.9.5.1. Starting Configuration



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 1  
Bridge.2 = 2  
ACCEPTABLE\_L1\_LATENCY.default = 3  
L1\_EXIT\_LATENCY.default = 0  
ASPML1Enable = 2

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 1  
Bridge.2 = 3  
ACCEPTABLE\_L1\_LATENCY.default = 3  
L1\_EXIT\_LATENCY.default = 0  
ASPML1Enable = 2

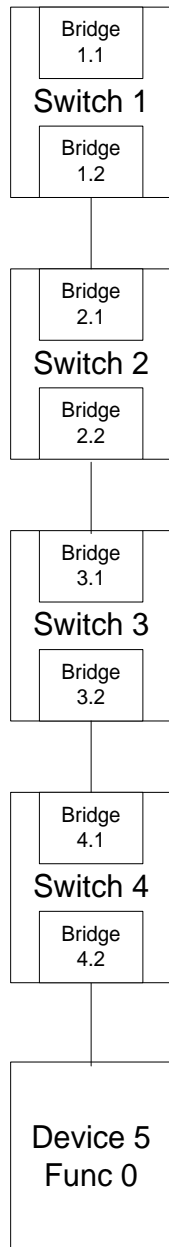
**[Device 3 Function 0]**

**DeviceType** = PCIExpressEndPoint  
VENDORID = 0x8086  
DEVICEID = 0x6001  
NumberPorts = 1  
Bridge.2 = 2  
ACCEPTABLE\_L1\_LATENCY.default = 1  
L1\_EXIT\_LATENCY.default = 0  
ASPML1Enable = 0



### 3.9.6. Test 9.6: Four Switch Delays Cause Potential L1 Mismatch - Informational Only Test

#### 3.9.6.1. Starting Configuration



**[Device 1 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 2  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 2 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 3  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 3 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 4  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 4 Function 0]**

**DeviceType** = PCIExpressSwitchSlots  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 NumberPorts = 1  
 Bridge.2 = 5  
 ACCEPTABLE\_L1\_LATENCY.default = 3  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 2

**[Device 5 Function 0]**

**DeviceType** = PCIExpressEndPoint  
 VENDORID = 0x8086  
 DEVICEID = 0x6001  
 ACCEPTABLE\_L1\_LATENCY.default = 2  
 L1\_EXIT\_LATENCY.default = 0  
 ASPML1Enable = 0

## **3.10. Test 10: Worst Case Device Response Time to Configuration Requests After Reset - Informational Only Test**

This test is run on downstream ports exposed by the platform under test. A PCI Express Test Device is used on the port under test. The device is programmed to respond to configuration cycles with worst case allowable response time. The test verifies that the system is still able to configure the platform test device.

### **3.10.1. Starting Configuration**

The Platform Test Device is connected to the downstream port under test.

### **3.10.2. Overview of Test Steps**

The test software performs the following steps.

1. Test software prompts the user to connect the platform test device to the slot to be tested.
2. Test software configures the platform test device to respond to the first configuration cycle (after the next system reset) 999 milliseconds (+0%/-5%) after the end of reset. The test device is also programmed to record the length of time from the end of reset to the arrival of the first configuration request.
3. The system under test is reset.
4. The platform test device starts a timer at the end of reset.
5. The test device records the arrival time of the first configuration request after reset. It waits until the time indicated in step 2 to respond to the request.
6. The test is repeated with the platform test device configured to respond immediately to the first configuration access.
7. The test is repeated with the platform test device configured to respond immediately to the first configuration access after reset.
8. The test is repeated for each available slot on the platform under test.

### 3.10.3. Results Interpretation

The test transcribes all results to a text based log file.

- ❑ The test fails if the platform test device is not configured correctly by the system when configured to respond to the first configuration cycle slightly before the 1 second system timeout.
- ❑ The test fails if the platform test device is not configured correctly by the system when configured to respond immediately to the first configuration cycle after reset.
- ❑ The test fails if the platform sends the first configuration request to the platform test device less than 100 milliseconds from the end of reset in any case.

## 3.11. Test 11 - Worst Case Device Entry into Link Training After Reset - Informational Only Test

This test is run on downstream ports exposed by the platform under test. A PCI Express Platform Test Device is used on the port under test. The device is programmed to enter the active state for link training in the worst case allowable time after reset. The test verifies that the system is still able to configure the platform test device.

### 3.11.1. Starting Configuration

The Test Device is connected to the downstream port under test.

### 3.11.2. Overview of Test Steps

The test software performs the following steps.

1. Test software prompts the user to connect the platform test device to the slot to be tested
2. Test software configures the platform test device to enter the initial active link training state (after the next system reset) 20 milliseconds (+0%/-5%) after the end of reset. The test device is also programmed to record the length of time from the end of reset to the arrival of the first configuration request (if supported by the platform test device).
3. The system under test is reset.
4. The platform test device starts a timer at the end of reset.
5. The test is repeated with the platform test device configured to enter the initial active link training state directly (as quickly as possible) after the end of reset.
6. The test is repeated with the platform test device configured to enter initial link training state immediately after reset.
7. The test is repeated for each available slot on the platform under test.

### 3.11.2.1. Results Interpretation

- ❑ The test transcribes all results to a text based log file.
- ❑ The test fails if the platform test device is not configured correctly by the system when configured to enter the initial active link training state slightly before the 20 millisecond requirement.
- ❑ The test fails if the platform test device is not configured correctly by the system when configured to enter the initial active link training state immediately after reset.
- ❑ The test fails if the platform sends the first configuration request to the platform test device less than 100 milliseconds from the end of reset in any case.

## 3.12. Test 12: Platform Response to Configuration Retry Responses - Informational Only Test

This test is run on downstream ports exposed by the platform under test. A PCI Express Platform Test Device is used on the port under test. The device is programmed to use the configuration retry response to the first configuration cycle following a reset and to other configuration cycles throughout the configuration process.

### 3.12.1. Starting Configuration

The Platform Test Device is connected to the downstream port under test.

### 3.12.2. Overview of Test Steps

The test software performs the following steps.

1. Test software prompts the user to connect the platform test device to the slot to be tested.
2. Test software configures the platform test device to respond to the first configuration cycle (after the next system reset) 999 milliseconds (+0%/-5%) after the end of reset. The test device is also programmed to record the length of time from the end of reset to the arrival of the first configuration request. The device is also configured to respond (at 999 milliseconds) with a Configuration Retry Status response. After the cycle is re-issued it responds normally to subsequent requests.
3. The system under test is reset.
4. The platform test device starts a timer at the end of reset.
5. The test device records the arrival time of the first configuration request after reset. It waits until the time indicated in step 2 to respond to the request with a Configuration Retry Status.

6. The test is repeated with the platform test device configured to respond immediately to the first configuration access after reset with a Configuration Retry Status. Subsequent requests are handled normally.
7. The test is repeated (steps 2-6) with the platform test device configured to respond initially to every configuration access with a Configuration Retry Status. When the initial request is retried it responds normally to all cycles.
8. The test is repeated for each available slot on the platform under test.

### 3.12.3. Results Interpretation

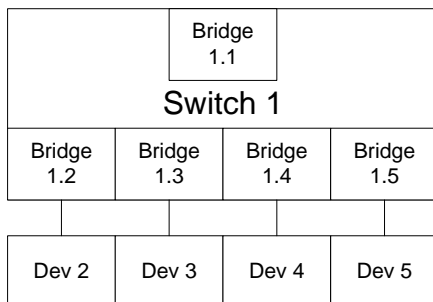
- ❑ The test transcribes all results to a text based log file.
- ❑ The test fails if the platform test device is not configured correctly by the system when configured to respond to the first configuration cycle slightly before the 1 second system timeout with a Configuration Retry Status.
- ❑ The test fails if the platform test device is not configured correctly by the system when configured to respond immediately to the first configuration cycle after reset with a Configuration Retry Status.
- ❑ The test fails if the platform sends the first configuration request to the platform test device less than 100 milliseconds from the end of reset in any case.
- ❑ The test fails if the platform test device is not configured correctly when it responds initially to all configuration requests with a Configuration Retry Status.

### 3.13. Test 13: Platform Response to “Broken” Device during Configuration - Informational Only Test

This test is run on downstream ports exposed by the platform under test. A PCI Express Platform Test Device(s) is used on the port under test. The device(s) is configured to represent a topology with multiple devices under a switch. The first device does not respond to configuration cycles, and after 1 second should be treated by the system as a broken device. The test verifies that the platform correctly treats the first device as a broken device and still configures the remaining devices behind the switch.

#### 3.13.1. Starting Configuration

The test device topology used in testing is shown on the following drawing.



##### [Device 1 Function 0]

```

DeviceType = PCIExpressSwitchSlots
VENDORID = 0x8086
DEVICEID = 0x6001
NumberPorts = 4
Bridge.2 = 2
Bridge.3 = 3
Bridge.4 = 4
Bridge.5 = 5
  
```

##### [Device 2 Function 0]

```

DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
  
```

```

Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0
NoResponse = 1

[Device 3 Function 0]
DeviceType = PCIExpressLegacyEndPoint
VENDORID = 0x8086
DEVICEID = 0x6001
Bar1.type = Mem32
Bar1.size = 64kb
Bar2.type = IO
Bar2.size = 16b
Bar3.type = Mem32
Bar3.size = 0b
Bar4.type = Mem32
Bar4.size = 0b
Bar5.type = Mem32
Bar5.size = 0
Bar6.type = Mem32
Bar6.size = 0

[Device 4 Function 0]
DeviceType = Device 3 Function 0

[Device 5 Function 0]
DeviceType = Device 3 Function 0

```

### 3.13.2. Overview of Test Steps

1. The test software performs the following steps.
2. Test software prompts the user to connect the platform test device(s) to the slot to be tested.
3. Test software configures the platform test device(s) to act as the topology specified above.
4. The system under test is reset.
5. The platform test device starts a timer at the end of reset.
6. The test device(s) records the arrival time of the first configuration request after reset. It responds normally to configuration cycles targeted to the switch and the responsive test devices. It does not respond to configuration cycles targeted to the non-responsive device.
7. The test is repeated with the identical configuration, except the non-responsive device below the switch responds only to the first configuration request and not to any subsequent requests.



8. The test is repeated for each available slot on the platform under test.

### **3.13.3. Results Interpretation**

The test fails if any test described in Section 2.1.2 fails for responsive devices.

## **3.14. Tests 14.x: Bios/OS Interaction Standard Test Cases**

There is a 3.0 revision of the PCI Firmware Specification. There are plans to test at least two Firmware interface methods relation to PCI Express. Currently, the exact test descriptions for these cases are TBD.

### **3.14.1. Test 14.1: MMCFG Reporting Of Extended Config Access Base Address - Informational Only Test**

### **3.14.2. Test 14.2: OHSP Method for Hot Plug Capable Systems - Informational Only Test**

### 3.15. Test 15: Max\_Payload\_Size Test – Required Test

TXN.02.02#13	FW	02.02.02.00.00#06	Software must not set the Max_Payload_Size, in different functions of the same device, to different values (unless software is aware of an implementation specific situation).
--------------	----	-------------------	--

Test abstract –

No prerequisites.

Create a multi-function Endpoint (with 8 functions) and have each function advertise different values for Device\_Capabilities.Max\_Payload\_Size\_Supported.

Check that after SBIOS configuration of the device, the values of Device\_Control.Max\_Payload\_Size for each function in the device is the same, and does not exceed the value advertised by the function with the smallest advertised Max\_Payload\_Size\_Supported (and the value advertised by the Root Port). Check that the value of Max\_Payload\_Size for each port of the Switch is equal or larger than the smallest programmed value (of both Root Port and Endpoint function) on that port's link hierarchy. Check that the value of Max\_Payload\_Size for the Root Port is equal to the value of the smallest programmed value of all Endpoint functions.

### 3.16. Test 16: End-End TLP Prefix Test – Informational-only Test

TXN.02.22#28	FW	02.02.10.02.00#09	Software must not enable a Requester to send a TLP with an End-End TLP Prefix, to a receiving device that doesn't support End-End TLP Prefixes.
--------------	----	-------------------	---

Test abstract –

Prerequisite that the Root Port has Device\_Capabilities\_2.Extended\_Fmt\_Field\_Supported=1 and Device\_Capabilities\_2.End-End\_TLP\_Prefix\_Supported=1. Also Root Port has TPH Requester Capability structure with TPH\_Requester\_Capability.Extended\_TPH\_Requester\_Supported=1.

Create a hierarchy with a Switch and multiple Endpoint devices. All ports of the Switch must have Device\_Capabilities\_2.Extended\_Fmt\_Field\_Supported=1, Device\_Capabilities\_2.End-End\_TLP\_Prefix\_Supported=1, and Device\_Capabilities\_2.Max\_End-End\_TLP\_Prefixes=00b. For the Endpoints, some should have Extended\_Fmt\_Field\_Supported=1, End-End\_TLP\_Prefix\_Supported=1, Max\_End-End\_TLP\_Prefixes=various values, and have Device\_Capabilities\_2.TPH\_Completer\_Supported=11b, while others should have Extended\_Fmt\_Field\_Supported=0, End-End\_TLP\_Prefix\_Supported=0, Max\_End-

End\_TLP\_Prefixes=00b, and Device\_Capabilities\_2.TPH\_Completer\_Supported=00b. Some of the Endpoints should be multi-function, but all functions within the same device must have the same value for these fields.

Check that Root Port does not have TPH\_Requeseter\_Control.TPH\_Requester\_Enable=11b (ie. Allow Extended TPH).

### 3.17. Test 17: MFVC Capability Test – Required Test

TXN.05.01#06	FW	02.05.01.00.00#04	Software must ensure that VC ID assignment is the same for both ports on each side of the link. If one port contains a MFVC Capability structure, then this applies to the MFVC Capability structure on one side of the link and the VC Capability structure on the other side of the link. If neither port contains a MFVC Capability structure, then this applies to the VC Capability structure on both sides of the link.
--------------	----	-------------------	---

- ❑ Test abstract –  
Prerequisite that the Root Port has support for at least VC1.
- ❑ Repeat the VC Capability tests, but using a multi-function Endpoint that contains a MFVC Capability structure in Function 0, and some other functions contain a VC Capability Structure (Extended Capability ID = 0009h).
- ❑ Check that after SBIOS configuration if other VCs (other than VC0) are enabled, that they are done correctly both on the Upstream link (MFVC Capability structure), and within the multi-function Endpoint (VC Capability structure).

### 3.18. Test 18: VC Mapping Test – Required Test

TXN.05.02#01	FW	02.05.02.00.00#01	Every Traffic Class that is supported must be mapped to one of the Virtual Channels.
TXN.05.02#03	FW	02.05.02.00.00#02	One TC must not be mapped to multiple VCs in any port or Endpoint Function.

Test abstract –  
No prerequisites.

Create a link hierarchy with Switches and multi-function Endpoints that support a VC Capability structure on each port.

Check that after SBIOS configuration, on each port, each TC is mapped to one and only one VC (note: multiple TCs may be mapped to the same VC). Check that the TC to VC mapping on each end of the same link is identical.

### 3.19. Test 19: PME Enable Test – Required Test

PMG.03.13#07	FW	05.03.03.02.00#09	Software must enable all devices that participate in link wakeup, including those that propagate the Beacon signal, to consume Vaux power.
--------------	----	-------------------	--

Test abstract –

Prerequisite that the Root Port has support for PME.

Create a link hierarchy with Switches and multi-function Endpoints, that all support PME for D3-cold. They should support the Power Management Capability structure and request the minimum non-zero Aux power value in the Power\_Management\_Capability.Aux\_Current\_Field. They should all support the Device\_Control.Auxiliary\_Power\_PM\_Enable bit as RWS.

Check that if an Endpoint function has Power\_Management\_Control/Status.PME\_En=1, then the link path to the Root port also has PME\_En=1.

### 3.20. Test 20 Slot Power Test – Required Test

SYS.09.00#07	FW	06.09.00.00.00#02	Software must correctly program the Slot Power Limit Value and Slot Power Limit Scale fields of the Slot Capabilities registers of Downstream ports connected to slots to one of the maximum values specified for the form factor, based on the capability of the platform.
CEM.04#32	FW	CEM.04.02.00.00#08	System firmware must update the slot power limit for the add-in card prior to invoking the option ROM for that add-in card.

Test abstract –

No prerequisites.

Create a link hierarchy with an Endpoint function that has Link\_Capabilities.Maximum\_Link\_Width=010000b (ie. x16).

Check that after SBIOS configuration, but before Endpoint BIOS is posted, the Root port's Slot\_Capabilities.Slot\_Power\_Limit\_Value and Slot\_Capabilities\_Slot\_Power\_Limit\_Scale values indicate greater than 25W (if the Endpoint function has been enabled via the Command register).

### 3.21. Test 21 LTR Mechanism Test – Required Test

SYS.19.01#01	FW	06.18.00.00.00#01	If LTR (optional) is supported, software must not enable LTR in an Endpoint connected to a link hierarchy that contains either a Root Complex that doesn't support LTR or a Switch that doesn't support LTR.
--------------	----	-------------------	--

Test abstract –

Root Port must advertise Device\_Capabilities\_2.LTR\_Mechanism\_Supported=1.

Create a link hierarchy with multiple intermediate Switches connected to an Endpoint. The Endpoint should have Device\_Capabilities\_2.LTR\_Mechanism\_Supported=1 and implement a LTR Capability structure. All but one of the Switches should have Device\_Capabilities\_2.LTR\_Mechanism\_Supported=1 and a LTR Capability structure (upstream ports only), and one Switch should have Device\_Capabilities\_2.LTR\_Mechanism\_Supported=0 and no LTR Capability structure.

Check that Root Port Device\_Control\_2.LTR\_Mechanism\_Enable=0. Check that all Switch ports Device\_Control\_2.LTR\_Mechanism\_Enable=0. Check that Endpoint Device\_Control\_2.LTR\_Mechanism\_Enable=0.

### 3.22. Test 22 ASPM Optionality Test – Required Test

CFG.08.07#16	FW	07.08.07.00.00#03	Software can only enable ASPM L1 if both Upstream and Downstream components on the link support ASPM L1. Software must enable ASPM L1 on the Upstream port before enabling ASPM L1 on the Downstream port. Software must disable ASPM L1 on the Downstream port before disabling ASPM L1 on the Upstream port.
PMG.04.03#05	FW	E5.04.01.01.01#01a	Software must ensure that both sides of the link support L0s, before it enables L0s on either side of the link.

Test abstract –

No prerequisites.

Create a link hierarchy with a Switch and a non-ARI multi-function Endpoint device with 8 functions. Each test case will have the Switch ports and functions report different values in Link\_Capabilities.ASPM\_Support (00b, 01b, 10b, 11b). They will all report Link\_Capabilities.ASPM\_Optionality\_Compliance=1.

Check that for each segment of the link hierarchy (including the Root Port), that the value in Link\_Control.Active\_State\_Power\_Management\_Control is a value that corresponds to an ASPM setting that is supported by the port/function on either end of that link segment (a value is 00b is always acceptable in all cases).

## 3.23. Test 23: Common Clock Tests

### 3.23.1. Test 23a: Common Clock Test 1 – Required Test

CFG.08.07#34	FW	07.08.07.00.00#22	Software must write the same value to the Common Clock Configuration bit for all functions in a non-ARI multi-function device.
CFG.08.07#26	FW	07.08.07.00.00#46	Software must read the Slot Clock Configuration bit of the devices on both sides of the link to determine if Common Clock is supported and if so, program the Common Clock Configuration bit on the devices on both sides of the link to be 1.

Test abstract –  
No prerequisites.

Create a non-ARI multi-function Endpoint device with 8 functions. Each function has Link\_Status.Slot\_Clock\_Configuration=1.

After SBIOS configuration check the Downstream port of the link (the Root Port) that Link\_Control.Common\_Clock\_Configuration=1 (if Link\_Status.Slot\_Clock\_Configuration=0 on both ends of the link), or that Common\_Clock\_Configuration=0 (if Slot\_Clock\_Configuration=0 on both ends of the link). Check for the Upstream port of the link (each Endpoint function) that the Common\_Clock\_Configuration value is identical to the immediate Upstream component's (the Root Port) Common\_Clock\_Configuration value.

### 3.23.2. Test 23B: Common Clock Test 2 – Required Test

CFG.08.07#34	FW	07.08.07.00.00#22	Software must write the same value to the Common Clock Configuration bit for all functions in a non-ARI multi-function device.
CFG.08.07#26	FW	07.08.07.00.00#46	Software must read the Slot Clock Configuration bit of the devices on both sides of the link to determine if Common Clock is supported and if so, program the Common Clock Configuration bit on the devices on both sides of the link to be 1.

Test abstract –

No prerequisites.

Create a link hierarchy with a Switch and a non-ARI multi-function Endpoint device with 8 functions. Each function has Link\_Status.Slot\_Clock\_Configuration=1. For some test cases, the Switch Upstream Port has Slot\_Clock\_Configuration=1 and some have Slot\_Clock\_Configuration=0. For some test cases, the Switch Downstream Port has Slot\_Clock\_Configuration=1 and some have Slot\_Clock\_Configuration=0.

After SBIOS configuration check each Downstream port of each link (including the Root Port) that Link\_Control.Common\_Clock\_Configuration=1 (if Link\_Status.Slot\_Clock\_Configuration=0 on both ends of the link), or that Common\_Clock\_Configuration=0 (if Slot\_Clock\_Configuration=0 on both ends of the link). Check for each Upstream port of each link that the Common\_Clock\_Configuration value is identical to the immediate Upstream component's Common\_Clock\_Configuration value.

## 3.24. Test 24: Clock Power Management Tests

### 3.24.1. Test 24a: Clock Power Management Test 1 – Required Test

CFG.08.07#35	FW	07.08.07.00.00#32	Software must only set the Enable Clock Power Management bit to 1 in any function of a non-ARI multi-function device, if all functions in the non-ARI multi-function device return a 1 in the Clock Power Management bit.
--------------	----	-------------------	---

Test abstract –  
No prerequisites.

Create a non-ARI multi-function Endpoint device with 8 functions. For each test case, make most functions report `Link_Capabilities.Clock_Power_Management=1`, but at least one function must report `Clock_Power_Management=0`.

Check that `Link_Control.Enable_Clock_Power_Management=0` in all Endpoint functions.

### 3.24.2. Test 24b: Clock Power Management Test 2 – Required Test

CFG.08.07#35	FW	07.08.07.00.00#32	Software must only set the Enable Clock Power Management bit to 1 in any function of a non-ARI multi-function device, if all functions in the non-ARI multi-function device return a 1 in the Clock Power Management bit.
--------------	----	-------------------	---

Test abstract –  
No prerequisites.

Create a link hierarchy with a Switch and a non-ARI multi-function Endpoint device with 8 functions. For each test case, make most functions report

`Link_Capabilities.Clock_Power_Management=1`, but at least one function must report `Clock_Power_Management=0`. For some test cases, the Switch Downstream Port has `Clock_Power_Management=1` and some have `Clock_Power_Management=0`.

Check that `Link_Control.Enable_Clock_Power_Management=0` in all Upstream ports (Switch and Endpoint functions).



## 3.25. Test 25: Resizable BAR Test – Required Test

Test abstract –  
No prerequisites.

Create a non-ARI multi-function Endpoint device with 8 functions. Each function should have all 5 Memory BARs implemented. For some test cases, the Function will have 32-bit BARs, while other test cases, it will have 64-bit BARs (and therefore have less than 5 BARs). Each function shall implement the Resizable BAR Capability structure, and for each implemented Memory BAR will have the Resizable\_BAR\_Capability register have bits 23-4 all set to 1 (supports all possible sizes from 1MB to 512GB).

After SBIOS configuration, check that BARs in each function has been programmed correctly (ie. have a non-conflicting and non-zero value). The SBIOS may use either the default BAR size, or it may use the Resizable BAR mechanism to set the BAR size prior to allocating resources for the function.

## 3.26. Test 26: OBFF Enable Test – Required Test

SYS.19.00#12	FW	E6.19.00.00.00#12 a	Software must not enable OBFF in a function, unless the hierarchy supports delivering OBFF to that function.
SYS.19.00#13	FW	E6.19.00.00.00#13 a	Software must configure the hierarchy (if it supports OBFF) so that either OBFF is delivered using WAKE# on those parts of the hierarchy where WAKE# interconnect is present, or OBFF is delivered using OBFF Messages on those parts of the hierarchy where OBFF Messages are supported.

Test abstract –  
No prerequisites.

Create a link hierarchy with a Switch and a non-ARI multi-function Endpoint device with 8 functions. The test cases will have the Switch ports and Endpoint function return different combinations in Device\_Capabilities\_2.OBFF\_Supported (ie. 00b, 01b, 10b, 11b) for different parts on the hierarchy.

After SBIOS configuration, check that if the Root Port reports Device\_Capabilities\_2.OBFF\_Supported=00b (not supported), then all ports (including the Root Port) and functions in the hierarchy have Device\_Control\_2.OBFF\_Enable=00b (disabled). Check if the Root if the Root Port reports OBFF\_Supported=non-zero (ie. some form of OBFF supported), but some port or function in the link hierarchy reports OBFF\_Supported=00b (not supported), that all ports (including the Root Port) and functions in the hierarchy have Device\_Control\_2.OBFF\_Enable=00b (disabled). Check if the Root if the Root Port reports OBFF\_Supported=non-zero (some form of OBFF supported), and all ports and functions in the link hierarchy reports OBFF\_Supported=non-zero (some form of OBFF supported), that all ports

(including the Root Port) and functions in the hierarchy either have Device\_Control\_2.OBFF\_Enable=00b (disabled), or have Device\_Control\_2.OBFF\_Enable return a value that corresponds to a supported OBFF delivery method supported by that part of the link.

## 3.27. Test 27: Various Extended Capabilities Present Test – Required Test

No prerequisites.

Create a non-ARI multi-function Endpoint device with 8 functions. Each function should have the following Capability structures (with valid information):

- ☐ Advanced Error Reporting Capability (the reporting interacts with some of the other capabilities)
- ☐ Device Serial Number Capability
- ☐ Power Budgeting Capability (indicating either less than 25 W total power, or System\_Allocated=1)
- ☐ Vendor-Specific Capability
- ☐ ACS Capability
- ☐ DPA Capability
- ☐ Multicast Capability
- ☐ Protocol Multiplexing Capability

After SBIOS configuration, check that the normal configuration has occurred, and that the presence of these Capability structures has not prevented BAR allocation, or BIOS posting.

## 3.28. Test 28: Various IOV Capabilities Present Test – Required Test

No prerequisites.

Create a non-ARI multi-function Endpoint device with 8 functions. Each function should have the following Capability structures (with valid information):

- ☐ Advanced Error Reporting Capability (the reporting interacts with some of the other capabilities)
- ☐ ATS Capability
- ☐ PRI Capability
- ☐ SR-IOV Capability
- ☐ MR-IOV Capability

After SBIOS configuration, check that the normal configuration has occurred, and that the presence of these Capability structures has not prevented BAR allocation, or BIOS posting.

## 3.29. Test 29: ARI Configuration Test – Required Test

No prerequisites.

Create a link hierarchy with a Switch and a multi-function Endpoint device with 8 functions. Each test case will have different combinations of ARI settings for Downstream ports, so that most ports will report `Device_Capabilities_2.ARI_Forwarding_Supported=1`, but some will report `Device_Capabilities_2.ARI_Forwarding_Supported=0`. For Upstream ports and functions, most will implement an ARI Capability, but some will not.

After SBIOS configuration, check in all Downstream ports (including the Root Port) that `Device_Control_2.ARI_Forwarding_Enable=0`. For Upstream ports and Function 0 only, `ARI_Control.MFVC_Function_Groups_Enable=0` and `ACS_Function_Groups_Enable=0`.

# PCI Express 1.1 Register and Capability - Appendix A: Defaults, Contents, and Characteristics for Functions in Test Cases

## A.1. Type 0 Configuration Header

### A.1.1. Standard Upstream Port Device Type 0 Configuration Header

#### A.1.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) – DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance

#### A.1.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE

BIT	ACCESS	DEFAULT	DESCRIPTION
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

### A.1.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6	RO	0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	00b	DEVSEL TIMING
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

### A.1.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
4:0	RO	00000b	MINOR REVISION ID
7:5	RO	001b	MAJOR REVISION ID

### A.1.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	PROG I/F CODE
15:8	RO	00h	SUB-CLASS CODE
23:16	RO	FFh	CLASS CODE

### A.1.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	CACHE LINE SIZE

### A.1.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	LATENCY TIMER

### A.1.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
6:0	RO	0000000b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

### A.1.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED

### A.1.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h)

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

#### A.1.1.11 Default IO BAR – DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

#### A.1.1.12 Default 32 bit Memory BAR – DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
3	RO	0	Prefetchable.
31:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**A.1.1.13 Default 64 bit Memory BAR – 2 DWORDs**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**A.1.1.14 PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	CARDBUS INFORMATION STRUCTURE POINTER

**A.1.1.15 PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0000h	SUBSYSTEM VENDOR ID

**A.1.1.16 PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0000h	SUBSYSTEM ID

**A.1.1.17 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**A.1.1.18 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

**A.1.1.19 PCI RESERVED REGISTER (OFFSET 38h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**A.1.1.20 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Interrupt Line

**A.1.1.21 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	01h	Interrupt Pin (INTA supported)

**A.1.1.22 PCI MIN\_GNT/MAX\_LAT REGISTERS (OFFSET 3Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED



## A.2. Type 1 Configuration Header

### A.2.1. Standard Bridge Device Type 1 Configuration Header

#### A.2.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance

#### A.2.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

#### A.2.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6		0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	0	DEVSEL TIMINGS
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

#### A.2.1.4 PCI REVISION ID REGISTER (Offset 08h) - BYTE

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	0	MINOR REVISION ID
7:5	RO	1	MAJOR REVISION ID

#### A.2.1.5 PCI CLASS CODE REGISTER (Offset 09h) – 3 BYTEs

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PROG I/F CODE
15:8	RO	04h	SUB-CLASS CODE
23:16	RO	06h	CLASS CODE

#### A.2.1.6 PCI CACHE LINE SIZE REGISTER (Offset 0Ch) – BYTE

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	0	CACHE LINE SIZE

#### A.2.1.7 PCI MASTER LATENCY TIMER REGISTER (Offset 0Dh) – BYTE

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PRIMARY LATENCY TIMER

**A.2.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
6:0	RO	0000001b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

**A.2.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED

**A.2.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h)**

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

**A.2.1.10.1. Default IO BAR - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

## A.2.1.10.2. Default 32 bit Memory BAR - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
4	RO	0	Prefetchable.
31:5	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

## A.2.1.10.3. Default 64 bit Memory BAR – 2 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:5	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

## A.2.1.11 PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Primary Bus Number

## A.2.1.12 PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Secondary Bus Number

## A.2.1.13 PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Subordinate Bus Number

### A.2.1.14 PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	Secondary Latency Timer

### A.2.1.15 PCI I/O BASE REGISTER

#### A.2.1.15.1. I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Base Address 15:12

#### A.2.1.15.2. I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Base Address 15:12

### A.2.1.16 PCI I/O Limit REGISTER

#### A.2.1.16.1. I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	00h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Limit Address 15:12

#### A.2.1.16.2. I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Limit Address 15:12

### A.2.1.17 PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
4:0	RO	0	Reserved
5	RO	0	66 MHz Capable
6	RO	0	Reserved
7	RO	0	Fast Back-to-Back Transactions Capable
8	RW1C	0	Master Data Parity Error
10:9	RO	00b	DEVSEL Timing
11	RW1C	0	Signaled Target Abort
12	RW1C	0	Received Target Abort
13	RW1C	0	Received Master Abort
14	RW1C	0	Received System Error
15	RW1C	0	Detected Parity Error

### A.2.1.18 PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0	Reserved
15:4	RW	0	Memory Base Address Bits 31:20

### A.2.1.19 PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0	Reserved
15:4	RW	0	Memory Limit Address Bits 31:20

### A.2.1.20 PCI PREFETCHABLE MEMORY BASE

#### A.2.1.20.1. PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

#### A.2.1.20.2. PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

### A.2.1.21 PCI PREFETCHABLE MEMORY LIMIT

#### A.2.1.21.1. PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

#### A.2.1.21.2. PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

### A.2.1.22 PCI PREFETCHABLE MEMORY BASE UPPER 32

#### A.2.1.22.1. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Base Address Upper 32 Bits

#### A.2.1.22.2. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Base Address Upper 32 Bits

### A.2.1.23 PCI PREFETCHABLE MEMORY LIMIT UPPER 32

#### A.2.1.23.1. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Limit Address Upper 32 Bits

#### A.2.1.23.2. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Limit Address Upper 32 Bits

### A.2.1.24 PCI I/O BASE UPPER 16

#### A.2.1.24.1. I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	I/O Base Address Upper 16 Bits

#### A.2.1.24.2. I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0	I/O Base Address Upper 16 Bits

### A.2.1.25 PCI I/O LIMIT UPPER 16

#### A.2.1.25.1. I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	I/O Limit Address Upper 16 Bits



**A.2.1.25.2. I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD**

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0	I/O Limit Address Upper 16 Bits

**A.2.1.26 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD**

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

**A.2.1.27 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD**

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**A.2.1.28 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE**

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Interrupt Line

**A.2.1.29 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE**

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	Interrupt Pin (INTA supported)

### A.2.1.30 PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Parity Error Response Enable
1	RW	0	SERR# Enable
2	RW	0	ISA Enable
3	RW	0	VGA Enable
4	RW	0	VGA 16-bit decode
5	RO	0	Master Abort Mode
6	RW	0	Secondary Bus Reset
7	RO	0	Fast Back-to-Back Transactions Enable
8	RO	0	Primary Discard Timer
9	RO	0	Secondary Discard Timer
10	RO	0	Discard Timer Status
11	RO	0	Discard Timer SERR# Enable
15:12	RO	0	Reserved

## A.3. Power Management Capability

### A.3.1. Standard Type 0 Header Device Power Management Capability Structure

#### A.3.1.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### A.3.1.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	010b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

### A.3.1.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
7:2	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### A.3.1.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### A.3.1.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## A.3.2. Standard Type 1 Header Device Power Management Capability Structure

### A.3.2.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER FIELDS (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### A.3.2.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	010b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

### A.3.2.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
7:2	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### A.3.2.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### A.3.2.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## A.4. MSI Capability Structures

### A.4.1. Standard Device PCI MSI-32 CAPABILITY STRUCTURE

#### A.4.1.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### A.4.1.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	0	64-BIT ADDRESS CAPABLE
15:8	RO	0	RESERVED

### A.4.1.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	32-BIT ADDRESS

### A.4.1.4 MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

## A.4.2. Standard Device PCI MSI-64 CAPABILITY STRUCTURE

### A.4.2.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### A.4.2.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	1	64-BIT ADDRESS CAPABLE
15:8	RO	0	RESERVED

### A.4.2.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	LOWER 32-BIT ADDRESS

### A.4.2.4 MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	UPPER 32-BIT ADDRESS

### A.4.2.5 MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

## A.5. PCI Express Capability Structure

### A.5.1. Standard Upstream Port Device PCI Express Capability Structure

#### A.5.1.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

### A.5.1.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1h	CAPABILITY VERSION
7:4	RO	0001b	DEVICE/PORT TYPE (Legacy PCI Express Endpoint. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED
13:9	RO	0	INTERRUPT MESSAGE NUMBER
15:14	RO	0	RESERVED

### A.5.1.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	CAPTURED SLOT POWER LIMIT VALUE
27:26	RO	0	CAPTURED SLOT POWER LIMIT SCALE
31:28	RO	0	RESERVED



**A.5.1.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	RESERVED

**A.5.1.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Set to 1 when non-posted requests are not yet completed and clear when they are completed)
15:6	RO	0	RESERVED

**A.5.1.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0001b	MAXIMUM LINK SPEED
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	0	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE
23:21	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER

**A.5.1.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RO	0	LINK DISABLE
5	RO	0	RETRAIN LINK
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
15:9	RO	0	RESERVED

**A.5.1.8 LINK STATUS REGISTER (OFFSET 12h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0001b	LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	0	DATA LINK LAYER LINK ACTIVE
15:14	RO	0	RESERVED

## A.5.2. Standard Downstream Port Device PCI Express Capability Structure

### A.5.2.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

### A.5.2.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1h	CAPABILITY VERSION
7:4	RO	0110b	DEVICE/PORT TYPE (Downstream Port of PCI Express Switch. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED (Other value will be noted in individual test cases)
13:9	RO	0	INTERRUPT MESSAGE NUMBER
15:14	RO	0	RESERVED

### A.5.2.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	RESERVED (Captured Slot Power Limit Value for Upstream Ports)
27:26	RO	0	RESERVED (Captured Slot Power Limit Scale for Upstream Ports)
31:28	RO	0	RESERVED

**A.5.2.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	RESERVED

**A.5.2.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Not used by default downstream device)
15:6	RO	0	RESERVED

**A.5.2.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0001b	MAXIMUM LINK SPEED
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	0	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE
23:21	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER

**A.5.2.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RW	0	LINK DISABLE
5	RO	0	RETRAIN LINK (always returns 0)
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
15:9	RO	0	RESERVED

### A.5.2.8 LINK STATUS REGISTER (OFFSET 12h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0001b	LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	0	DATA LINK LAYER LINK ACTIVE
15:14	RO	0	RESERVED

### A.5.3. Standard Downstream Port Device with Slot PCI Express Capability Structure

The registers for a default downstream port device with a slot are identical to those for a default downstream device without a slot from offset 00h through 13h inclusive, with the following exceptions:

#### A.5.3.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
8	RO	1	SLOT IMPLEMENTED

Also, the following additional registers to the PCI Express Capability structure are implemented with the indicated characteristics.

### A.5.3.2 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Attention Button Present
1	RO	0	Power Controller Present
2	RO	0	MRL Sensor Present
3	RO	0	Attention Indicator Present
4	RO	0	Power Indicator Present
5	RO	0	Hot-Plug Surprise
6	RO	0	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale
17	RO	0	Electromechanical Interlock Present
18	RO	0	No Command Completed Support
31:19	RO	0	Physical Slot Number

### A.5.3.3 SLOT CONTROL REGISTER (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Attention Button Pressed Enable
1	RO	0	Power Fault Detected Enable
2	RO	0	MRL Sensor Changed Enable
3	RO	0	Presence Detect Changed Enable
4	RO	0	Command Completed Interrupt Enable
5	RO	0	Hot-Plug Interrupt Enable
7:6	RW	00b	Attention Indicator Control
9:8	RW	00b	Power Indicator Control
10	RO	0	Power Controller Control (No Power Controller Implemented By Default)
11	RO	0	Electromechanical Interlock Control
12	RO	0	Data Link Layer State Changed Enable
15:13	RO	0	RESERVED



### A.5.3.4 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RW1C	0	Command Completed
5	RO	0	MRL Sensor State
6	RO	1	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status
8	RW1C	0	Data Link Layer State Changed
15:9	RO	0	RESERVED

### A.5.4. Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure

The registers for a default downstream port device with a hot plug capable slot are identical to those for a default downstream device without a slot from offset 00h through 13h inclusive, with the following exceptions:

#### A.5.4.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
8	RO	1	SLOT IMPLEMENTED

Also, the following additional registers to the PCI Express Capability structure are implemented with the indicated characteristics.

### A.5.4.2 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	Attention Button Present
1	RO	1	Power Controller Present
2	RO	1	MRL Sensor Present
3	RO	1	Attention Indicator Present
4	RO	1	Power Indicator Present
5	RO	1	Hot-Plug Surprise
6	RO	1	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale
17	RO	1	Electromechanical Interlock Present
18	RO	0	No Command Completed Support
31:19	RO	0	Physical Slot Number

### A.5.4.3 SLOT CONTROL REGISTER (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Attention Button Pressed Enable
1	RW	0	Power Fault Detected Enable
2	RW	0	MRL Sensor Changed Enable
3	RW	0	Presence Detect Changed Enable
4	RW	0	Command Completed Interrupt Enable
5	RW	0	Hot-Plug Interrupt Enable
7:6	RW	11b	Attention Indicator Control (Off)
9:8	RW	01b	Power Indicator Control (On)
10	RW	0	Power Controller Control
11	RO	0	Electromechanical Interlock Control
12	RW	0	Data Link Layer State Changed Enable
15:13	RO	0	RESERVED

#### A.5.4.4 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RW1C	0	Command Completed
5	RO	1	MRL Sensor State (Open)
6	RO	0	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status (Disengaged)
8	RW1C	0	Data Link Layer State Changed
15:9	RO	0	RESERVED

### A.6. PCI Express Advanced Error Reporting Capability

#### A.6.1. Standard PCI Express Advanced Error Reporting Capability

##### A.6.1.1 PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0001h	AER EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	138h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)

### A.6.1.2 UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED(optional)
3:1	RO	0	RESERVED
4	RW1CS	0	DATA LINK PROTOCOL ERROR STATUS
5	RW1CS	0	SURPRISE DOWN ERROR STATUS (optional)
11:6	RO	0	RESERVED
12	RW1CS	0	POISONED TLP STATUS
13	RO	0	FLOW CONTROL PROTOCOL ERROR STATUS (optional)
14	RW1CS	0	COMPLETION TIMEOUT STATUS
15	RO	0	COMPLETER ABORT STATUS (optional)
16	RW1CS	0	UNEXPECTED COMPLETION STATUS
17	RO	0	RECEIVER OVERFLOW STATUS (optional)
18	RW1CS	0	MALFORMED TLP STATUS
19	RO	0	ECRC ERROR STATUS (optional)
20	RW1CS	0	UNSUPPORTED REQUEST ERROR STATUS
31:21	RO	0	RESERVED

### A.6.1.3 UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED (optional)
3:1	RO	0	RESERVED
4	RWS	0	DATA LINK PROTOCOL ERROR MASK
5	RWS	0	SURPRISE DOWN ERROR MASK (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP MASK
13	RO	0	FLOW CONTROL PROTOCOL ERROR MASK (optional)
14	RWS	0	COMPLETION TIMEOUT MASK
15	RO	0	COMPLETER ABORT MASK (optional)
16	RWS	0	UNEXPECTED COMPLETION MASK
17	RO	0	RECEIVER OVERFLOW MASK (optional)
18	RWS	0	MALFORMED TLP MASK
19	RO	0	ECRC ERROR MASK (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR MASK
31:21	RO	0	RESERVED

### A.6.1.4 UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED (optional)
3:1	RO	0	RESERVED
4	RWS	1	DATA LINK PROTOCOL ERROR SEVERITY
5	RWS	0	SURPRISE DOWN ERROR SEVERITY (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP SEVERITY
13	RO	0	FLOW CONTROL PROTOCOL ERROR SEVERITY (optional)
14	RWS	0	COMPLETION TIMEOUT ERROR SEVERITY
15	RO	0	COMPLETER ABORT ERROR SEVERITY (optional)
16	RWS	0	UNEXPECTED COMPLETION ERROR SEVERITY
17	RO	0	RECEIVER OVERFLOW ERROR SEVERITY (optional)
18	RWS	1	MALFORMED TLP SEVERITY
19	RO	0	ECRC ERROR SEVERITY (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR SEVERITY
31:21	RO	0	RESERVED

### A.6.1.5 CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR STATUS (optional)
5:1	RO	0	RESERVED
6	RW1CS	0	BAD TLP STATUS
7	RW1CS	0	BAD DLLP STATUS
8	RW1CS	0	REPLAY_NUM ROLLOVER STATUS
11:9	RO	0	RESERVED
12	RW1CS	0	REPLAY TIMER TIMEOUT STATUS
13	RW1CS	0	ADVISORY NON-FATAL ERROR STATUS
31:14	RO	0	RESERVED

### A.6.1.6 CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR MASK (optional)
5:1	RO	0	RESERVED
6	RWS	0	BAD TLP MASK
7	RWS	0	BAD DLLP MASK
8	RWS	0	REPLAY_NUM ROLLOVER MASK
11:9	RO	0	RESERVED
12	RWS	0	REPLAY TIMER TIMEOUT MASK
13	RWS	1	ADVISORY NON-FATAL ERROR MASK
31:14	RO	0	RESERVED

### A.6.1.7 ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
4:0	ROS	0	FIRST ERROR POINTER
5	RO	0	ECRC GENERATION CAPABLE
6	RWS	0	ECRC GENERATION ENABLE
7	RO	0	ECRC CHECK CAPABLE
8	RWS	0	ECRC CHECK ENABLE
31:9	RO	0	RESERVED

### A.6.1.8 HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
127:0	ROS	0	Header of TLP For First Error

## A.7. PCI Express Virtual Channel Capability

### A.7.1. Standard Endpoint PCI Express Virtual Channel Capability

#### A.7.1.1 PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0002h	VC EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	600h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)

#### A.7.1.2 PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	111b	Extended VC Count (8 VCs)
3	RO	0	Reserved
6:4	RO	100b	Low Priority Extended VC Count
7	RO	0	Reserved
9:8	RO	0	Reference Clock (Reserved For Endpoints)
11:10	RO	0	Port Arbitration Table Entry Size (Reserved For Endpoints)
31:12	RO	0	Reserved

#### A.7.1.3 PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00000010b	VC Arbitration Capability
23:8	RO	0	Reserved
31:24	RO	70h	VC Arbitration Table Offset



**A.7.1.4 PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Load VC Arbitration Table (always returns 0)
3:1	RO	001b	VC Arbitration Select
15:4	RO	0	Reserved

**A.7.1.5 PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	VC Arbitration Table Status
15:1	RO	0	Reserved

**A.7.1.6 VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.7 VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	TC/VC Map (0)
7:1	RO	1111111b	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RO	0	VC ID
30:27	RO	0	Reserved
31	RO	1	VC Enable

### A.7.1.8 Reserved Register (OFFSET 18h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### A.7.1.9 VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.10 VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined

15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

#### A.7.1.11 VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	1	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

#### A.7.1.12 Reserved Register (OFFSET 24h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

#### A.7.1.13 VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.14 VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.15 VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	2	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### A.7.1.16 Reserved Register (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### A.7.1.17 VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.18 VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.19 VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	3	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

**A.7.1.20 Reserved Register (OFFSET 3Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	Reserved

**A.7.1.21 VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**A.7.1.22 VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.23 VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	4	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### A.7.1.24 Reserved Register (OFFSET 48h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### A.7.1.25 VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.26 VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.27 VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	5	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### A.7.1.28 Reserved Register (OFFSET 54h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved



### A.7.1.29 VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.30 VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.31 VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	6	VC ID

30:27	RO	0	Reserved
31	RW	0	VC Enable

#### A.7.1.32 Reserved Register (OFFSET 60h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

#### A.7.1.33 VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

#### A.7.1.34 VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### A.7.1.35 VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	7	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### A.7.1.36 Reserved Register (OFFSET 6Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### A.7.1.37 VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### A.7.1.38 VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RW	0	VC ID Phase 0
3	RW	0	Reserved
...			
126:124	RW	0	VC ID Phase 31
127	RW	0	Reserved

## A.8. PCI Express Power Budgeting Capability

### A.8.1. Standard Device PCI Express Power Budgeting Capability

#### A.8.1.1 PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0004h	Power Budgeting Extended Capability ID
19:16	RO	1h	Capability Version
31:20	RO	000h	Next Capability Offset (Last capability in list)

#### A.8.1.2 DATA SELECT REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Data Select
31:8	RO	0	Reserved

#### A.8.1.3 DATA REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h*	Base Power (*value depends on Data Select value)
9:8	RO	00b	Data Scale
12:10	RO	000b	PM Sub State
14:13	RO	00b	PM State
17:15	RO	000b*	Type (*value depends on Data Select value)
20:18	RO	000b	Power Rail
31:21	RO	0	Reserved

### A.8.1.4 POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	System Allocated
31:1	RO	0	Reserved

The default power budgeting capability contains the following Data Select Index and Data pairs.

Data Select Index: 0

Base Power 25

Data Scale 00b (1.0)

PM Sub State 000b (Default)

PM State 00b (D0)

Type 111b (Max)

Power Rail 000b (12 Volts)

Data Select Index: 1

Base Power 20

Data Scale 00b (1.0)

PM Sub State 000b (Default)

PM State 00b (D0)

Type 011b (Sustained)

Power Rail 000b (12 Volts)

## Appendix B PCI Express 2.0 Register and Capability – Defaults, Contents, and Characteristics for Functions in Test Cases

### B.1. Type 0 Configuration Header

#### B.1.1. Standard Upstream Port Type 0 Configuration Header

##### B.1.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance

**B.1.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

**B.1.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6	RO	0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	00b	DEVSEL TIMING
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

**B.1.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	00000b	MINOR REVISION ID
7:5	RO	001b	MAJOR REVISION ID

**B.1.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PROG I/F CODE
15:8	RO	00h	SUB-CLASS CODE
23:16	RO	FFh	CLASS CODE

**B.1.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	CACHE LINE SIZE

**B.1.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	LATENCY TIMER

**B.1.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
6:0	RO	0000000b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

**B.1.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED



**B.1.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h)**

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

**B.1.1.10.1. Default IO BAR - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.1.1.10.2. Default 32 bit Memory BAR - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
3	RO	0	Prefetchable.
31:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.1.1.10.3. Default 64 bit Memory BAR – 2 DWORDs**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.1.1.11 PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	CARDBUS INFORMATION STRUCTURE POINTER

**B.1.1.12 PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0000h	SUBSYSTEM VENDOR ID

**B.1.1.13 PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0000h	SUBSYSTEM ID

**B.1.1.14 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.1.1.15 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

**B.1.1.16 PCI RESERVED REGISTER (OFFSET 38h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.1.1.17 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Interrupt Line

**B.1.1.18 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	01h	Interrupt Pin (INTA supported)

**B.1.1.19 PCI MIN\_GNT/MAX\_LAT REGISTERS (OFFSET 3Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.2. Type 1 Configuration Header****B.2.1. Standard Bridge Device Type 1 Configuration Header****B.2.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance

**B.2.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

**B.2.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6	RO	0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	00b	DEVSEL TIMING
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

**B.2.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	00000b	MINOR REVISION ID
7:5	RO	001b	MAJOR REVISION ID

**B.2.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PROG I/F CODE
15:8	RO	04h	SUB-CLASS CODE
23:16	RO	06h	CLASS CODE

**B.2.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	CACHE LINE SIZE

**B.2.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PRIMARY LATENCY TIMER

**B.2.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
6:0	RO	0000001b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

**B.2.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED

**B.2.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h)**

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

**B.2.1.10.1. Default IO BAR - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.2.1.10.2. Default 32 bit Memory BAR - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
3	RO	0	Prefetchable.
31:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.2.1.10.3. Default 64 bit Memory BAR – 2 DWORDs**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**B.2.1.11 PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Primary Bus Number

### B.2.1.12 PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Secondary Bus Number

### B.2.1.13 PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Subordinate Bus Number

### B.2.1.14 PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	Secondary Latency Timer

### B.2.1.15 PCI I/O BASE REGISTER

#### B.2.1.15.1. I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Base Address 15:12

#### B.2.1.15.2. I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Base Address 15:12

**B.2.1.16 PCI I/O Limit REGISTER****B.2.1.16.1. I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Limit Address 15:12

**B.2.1.16.2. I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Limit Address 15:12

**B.2.1.17 PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	0	Reserved
5	RO	0	66 MHz Capable
6	RO	0	Reserved
7	RO	0	Fast Back-to-Back Transactions Capable
8	RW1C	0	Master Data Parity Error
10:9	RO	00b	DEVSEL Timing
11	RW1C	0	Signaled Target Abort
12	RW1C	0	Received Target Abort
13	RW1C	0	Received Master Abort
14	RW1C	0	Received System Error
15	RW1C	0	Detected Parity Error

**B.2.1.18 PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0	Reserved
15:4	RW	0	Memory Base Address Bits 31:20



**B.2.1.19 PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0	Reserved
15:4	RW	0	Memory Limit Address Bits 31:20

**B.2.1.20 PCI PREFETCHABLE MEMORY BASE****B.2.1.20.1. PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

**B.2.1.20.2. PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

**B.2.1.21 PCI PREFETCHABLE MEMORY LIMIT****B.2.1.21.1. PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

### B.2.1.21.2. PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

### B.2.1.22 PCI PREFETCHABLE MEMORY BASE UPPER 32

#### B.2.1.22.1. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Base Address Upper 32 Bits

#### B.2.1.22.2. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Base Address Upper 32 Bits

### B.2.1.23 PCI PREFETCHABLE MEMORY LIMIT UPPER 32

#### B.2.1.23.1. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Limit Address Upper 32 Bits

#### B.2.1.23.2. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Limit Address Upper 32 Bits

**B.2.1.24 PCI I/O BASE UPPER 16****B.2.1.24.1. I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	I/O Base Address Upper 16 Bits

**B.2.1.24.2. I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RW	0	I/O Base Address Upper 16 Bits

**B.2.1.25 PCI I/O LIMIT UPPER 16****B.2.1.25.1. I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	I/O Limit Address Upper 16 Bits

**B.2.1.25.2. I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RW	0	I/O Limit Address Upper 16 Bits

**B.2.1.26 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

### B.2.1.27 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

### B.2.1.28 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Interrupt Line

### B.2.1.29 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	Interrupt Pin (INTA supported)

### B.2.1.30 PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Parity Error Response Enable
1	RW	0	SERR# Enable
2	RW	0	ISA Enable
3	RW	0	VGA Enable
4	RW	0	VGA 16-bit decode
5	RO	0	Master Abort Mode
6	RW	0	Secondary Bus Reset
7	RO	0	Fast Back-to-Back Transactions Enable
8	RO	0	Primary Discard Timer
9	RO	0	Secondary Discard Timer
10	RO	0	Discard Timer Status
11	RO	0	Discard Timer SERR# Enable
15:12	RO	0	Reserved

## B.3. Power Management Capability

### B.3.1. Standard Type 0 Header Device Power Management Capability Structure

#### B.3.1.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### B.3.1.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	011b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

#### B.3.1.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
2	RO	0	RESERVED
3	RO	0	NO SOFT RESET
7:4	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### B.3.1.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### B.3.1.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## B.3.2. Standard Type 1 Header Device Power Management Capability Structure

### B.3.2.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### B.3.2.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	011b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT

BIT	ACCESS	DEFAULT	DESCRIPTION
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

### B.3.2.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
2	RO	0	RESERVED
3	RO	0	NO SOFT RESET
7:4	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### B.3.2.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### B.3.2.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## B.4. MSI Capability Structures

### B.4.1. Standard Device PCI MSI-32 CAPABILITY STRUCTURE

#### B.4.1.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### B.4.1.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	0	64-BIT ADDRESS CAPABLE
8	RO	0	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

#### B.4.1.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	32-BIT ADDRESS



**B.4.1.4 MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RW	0000h	MESSAGE DATA

**B.4.2. Standard Device PCI MSI-64 CAPABILITY STRUCTURE****B.4.2.1 MSI CAPABILITY FIELDS (OFFSET 00h)**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER

**B.4.2.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	1	64-BIT ADDRESS CAPABLE
8	RO	0	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

**B.4.2.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0:1	RO	0	RESERVED
31:2	RW	0	LOWER 32-BIT ADDRESS

#### B.4.2.4 MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	UPPER 32-BIT ADDRESS

#### B.4.2.5 MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

#### B.4.2.6 Standard Device PCI MSI-PVM-32 CAPABILITY STRUCTURE

#### B.4.2.7 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### B.4.2.8 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	0	64-BIT ADDRESS CAPABLE
8	RO	1	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

### B.4.2.9 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	32-BIT ADDRESS

### B.4.2.10 MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

### B.4.2.11 MSI RESERVED REGISTER (OFFSET 0Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED

### B.4.2.12 MSI MASK BITS REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	MASK BITS

### B.4.2.13 MSI PENDING BITS REGISTER (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	PENDING BITS

### B.4.3. Standard Device PCI MSI-PVM-64 CAPABILITY STRUCTURE

#### B.4.3.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### B.4.3.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	1	64-BIT ADDRESS CAPABLE
8	RO	1	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

#### B.4.3.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	LOWER 32-BIT ADDRESS

### B.4.3.4 MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	UPPER 32-BIT ADDRESS

### B.4.3.5 MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

### B.4.3.6 MSI RESERVED REGISTER (OFFSET 0Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED

### B.4.3.7 MSI MASK BITS REGISTER (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	MASK BITS

### B.4.3.8 MSI PENDING BITS REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	PENDING BITS

## B.5. PCI Express Capability Structure

### B.5.1. Standard Upstream Port Device PCI Express Capability Structure

#### B.5.1.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

#### B.5.1.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	2h	CAPABILITY VERSION
7:4	RO	0001b	DEVICE/PORT TYPE (Legacy PCI Express Endpoint. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED
13:9	RO	0	INTERRUPT MESSAGE NUMBER
14	RO	0	UNDEFINED
15	RO	0	RESERVED

### B.5.1.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	CAPTURED SLOT POWER LIMIT VALUE
27:26	RO	0	CAPTURED SLOT POWER LIMIT SCALE
28	RO	0	FUNCTION LEVEL RESET CAPABILITY
31:29	RO	0	RESERVED

### B.5.1.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	INITIATE FUNCTION LEVEL RESET (always returns 0)

**B.5.1.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Set to 1 when non-posted requests are not yet completed and clear when they are completed)
15:6	RO	0	RESERVED

**B.5.1.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0010b	SUPPORTED LINK SPEEDS
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	0	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE
21	RO	0	LINK BANDWIDTH NOTIFICATION CAPABILITY
23:22	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER



**B.5.1.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RO	0	LINK DISABLE
5	RO	0	RETRAIN LINK
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
9	RO	0	HARDWARE AUTONOMOUS WIDTH DISABLE
10	RO	0	LINK BANDWIDTH MANAGEMENT INTERRUPT ENABLE
11	RO	0	LINK AUTONOMOUS BANDWIDTH INTERRUPT ENABLE
15:12	RO	0	RESERVED

**B.5.1.8 LINK STATUS REGISTER (OFFSET 12h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0010b	CURRENT LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	0	DATA LINK LAYER LINK ACTIVE
14	RO	0	LINK BANDWIDTH MANAGEMENT STATUS
15	RO	0	LINK AUTONOMOUS BANDWIDTH STATUS

**B.5.1.9 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.5.1.10 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.11 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.12 ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.13 ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.14 ROOT STATUS REGISTER (OFFSET 20h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.5.1.15 DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	Completion Timeout Ranges Supported
4	RO	0	Completion Timeout Disable Supported
31:5	RO	0	RESERVED

**B.5.1.16 DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	Completion Timeout Value
4	RO	0	Completion Timeout Disable
15:5	RO	0	RESERVED

**B.5.1.17 DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.18 LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.5.1.19 LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RWS*	0010b*	Target Link Speed (*Must be RO-Zero for non-zero function number)
4	RWS*	0	Enter Compliance (*Must be RO-Zero for non-zero function number)
5	RO	0	Hardware Autonomous Speed Disable
6	RO	0	Selectable De-emphasis
9:7	RWS*	000b	Transmit Margin (*Must be RO-Zero for non-zero function number)
10	RWS*	0	Enter Modified Compliance (*Must be RO-Zero for non-zero function number)
11	RWS*	0	Compliance SOS (*Must be RO-Zero for non-zero function number)
12	RWS*	0	Compliance De-emphasis (*Must be RO-Zero for non-zero function number)
15:13	RO	0	RESERVED

**B.5.1.20 LINK STATUS 2 REGISTER (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	Current De-emphasis Level (test card is -3.5dB)
15:1	RO	0	RESERVED

**B.5.1.21 SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.5.1.22 SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.1.23 SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

## B.5.2. Standard Downstream Port Device PCI Express Capability Structure

### B.5.2.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

### B.5.2.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	2h	CAPABILITY VERSION
7:4	RO	0110b	DEVICE/PORT TYPE (Downstream Port of PCI Express Switch. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED (Other value will be noted in individual test cases)
13:9	RO	0	INTERRUPT MESSAGE NUMBER
14	RO	0	UNDEFINED
15	RO	0	RESERVED

### B.5.2.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	RESERVED (Captured Slot Power Limit Value for Upstream Ports)
27:26	RO	0	RESERVED (Captured Slot Power Limit Scale for Upstream Ports)
31:28	RO	0	RESERVED

### B.5.2.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	RESERVED

**B.5.2.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Not used by default downstream device)
15:6	RO	0	RESERVED

**B.5.2.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0010b	SUPPORTED LINK SPEEDS
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	0	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE
21	RO	1	LINK BANDWIDTH NOTIFICATION CAPABILITY (must be 1 for 5.0 GT/s capable downstream port)
23:22	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER

**B.5.2.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RW	0	LINK DISABLE
5	RO	0	RETRAIN LINK (always returns 0)
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
9	RO	0	HARDWARE AUTONOMOUS WIDTH DISABLE
10	RW	0	LINK BANDWIDTH MANAGEMENT INTERRUPT ENABLE
11	RW	0	LINK AUTONOMOUS BANDWIDTH INTERRUPT ENABLE
15:12	RO	0	RESERVED

**B.5.2.8 LINK STATUS REGISTER (OFFSET 12h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0010b	CURRENT LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	0	DATA LINK LAYER LINK ACTIVE
14	RW1C	0	LINK BANDWIDTH MANAGEMENT STATUS
15	RW1C	0	LINK AUTONOMOUS BANDWIDTH STATUS

**B.5.2.9 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED



**B.5.2.10 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.2.11 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
5:0	RO	0	RESERVED
6	RO	1	Presence Detect State (If no slot present must be 1)
15:7	RO	0	RESERVED

**B.5.2.12 ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.2.13 ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.2.14 ROOT STATUS REGISTER (OFFSET 20h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

### B.5.2.15 DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Ranges Supported
4	RO	0	Completion Timeout Disable Supported
31:5	RO	0	RESERVED

### B.5.2.16 DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Value
4	RO	0	Completion Timeout Disable
15:5	RO	0	RESERVED

### B.5.2.17 DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED

### B.5.2.18 LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	RESERVED

**B.5.2.19 LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RWS	0010b	Target Link Speed
4	RWS	0	Enter Compliance
5	RO	0	Hardware Autonomous Speed Disable
6	RO	1	Selectable De-emphasis (test card is -3.5dB)
9:7	RWS	000b	Transmit Margin
10	RWS	0	Enter Modified Compliance
11	RWS	0	Compliance SOS
12	RWS	0	Compliance De-emphasis
15:13	RO	0	RESERVED

**B.5.2.20 LINK STATUS 2 REGISTER (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	Current De-emphasis Level (test card is -3.5dB)
15:1	RO	0	RESERVED

**B.5.2.21 SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**B.5.2.22 SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.2.23 SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**B.5.3. Standard Downstream Port Device with Slot PCI Express Capability Structure**

The registers for a default downstream port device with a slot are identical to those for a default downstream device without a slot from offset 00h through 3Bh inclusive, with the following exceptions:

**B.5.3.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
8	RO	1	SLOT IMPLEMENTED

**B.5.3.2 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Attention Button Present
1	RO	0	Power Controller Present
2	RO	0	MRL Sensor Present
3	RO	0	Attention Indicator Present
4	RO	0	Power Indicator Present
5	RO	0	Hot-Plug Surprise
6	RO	0	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale
17	RO	0	Electromechanical Interlock Present
18	RO	0	No Command Completed Support
31:19	RO	0	Physical Slot Number

**B.5.3.3 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Attention Button Pressed Enable
1	RO	0	Power Fault Detected Enable
2	RO	0	MRL Sensor Changed Enable
3	RO	0	Presence Detect Changed Enable
4	RO	0	Command Completed Interrupt Enable
5	RO	0	Hot-Plug Interrupt Enable
7:6	RW	00b	Attention Indicator Control
9:8	RW	00b	Power Indicator Control
10	RO	0	Power Controller Control (No Power Controller Implemented By Default)
11	RO	0	Electromechanical Interlock Control
12	RO	0	Data Link Layer State Changed Enable
15:13	RO	0	RESERVED

**B.5.3.4 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RW1C	0	Command Completed
5	RO	0	MRL Sensor State
6	RO	1	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status
8	RW1C	0	Data Link Layer State Changed
15:9	RO	0	RESERVED

### B.5.4. Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure

The registers for a default downstream port device with a hot plug capable slot are identical to those for a default downstream device without a slot from offset 00h through 3Bh inclusive, with the following exceptions:

#### B.5.4.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
8	RO	1	SLOT IMPLEMENTED

#### B.5.4.2 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
20	RO	1	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE (Hot-Plug Capable slot must set this bit to 1)

#### B.5.4.3 LINK STATUS REGISTER (OFFSET 12h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
13	RO	1	DATA LINK LAYER LINK ACTIVE (Hot-Plug Capable slot must support this bit; DL_Active state = 1)

#### B.5.4.4 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	Attention Button Present
1	RO	1	Power Controller Present
2	RO	1	MRL Sensor Present
3	RO	1	Attention Indicator Present
4	RO	1	Power Indicator Present
5	RO	1	Hot-Plug Surprise
6	RO	1	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale

17	RO	1	Electromechanical Interlock Present
18	RO	1	No Command Completed Support
31:19	RO	0	Physical Slot Number

#### B.5.4.5 SLOT CONTROL REGISTER (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Attention Button Pressed Enable
1	RW	0	Power Fault Detected Enable
2	RW	0	MRL Sensor Changed Enable
3	RW	0	Presence Detect Changed Enable
4	RW	0	Command Completed Interrupt Enable
5	RW	0	Hot-Plug Interrupt Enable
7:6	RW	11b	Attention Indicator Control (Off)
9:8	RW	01b	Power Indicator Control (On)
10	RW	0	Power Controller Control
11	RO	0	Electromechanical Interlock Control
12	RW	0	Data Link Layer State Changed Enable (Hot-Plug Capable slot must support this bit)
15:13	RO	0	RESERVED

#### B.5.4.6 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RW1C	0	Command Completed
5	RO	1	MRL Sensor State (Open)
6	RO	0	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status (Disengaged)
8	RW1C	0	Data Link Layer State Changed (Hot-Plug Capable slot must support this bit)
15:9	RO	0	RESERVED

## B.6. PCI Express Advanced Error Reporting Capability

### B.6.1. Standard PCI Express Advanced Error Reporting Capability

#### B.6.1.1 PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0001h	AER EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	138h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)



## B.6.1.2 UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RW1CS	0	DATA LINK PROTOCOL ERROR STATUS
5	RO	0	SURPRISE DOWN ERROR STATUS (optional)
11:6	RO	0	RESERVED
12	RW1CS	0	POISONED TLP STATUS
13	RO	0	FLOW CONTROL PROTOCOL ERROR STATUS (optional)
14	RW1CS	0	COMPLETION TIMEOUT STATUS
15	RO	0	COMPLETER ABORT STATUS (optional)
16	RW1CS	0	UNEXPECTED COMPLETION STATUS
17	RO	0	RECEIVER OVERFLOW STATUS (optional)
18	RW1CS	0	MALFORMED TLP STATUS
19	RO	0	ECRC ERROR STATUS (optional)
20	RW1CS	0	UNSUPPORTED REQUEST ERROR STATUS
21	RO	0	ACS VIOLATION STATUS (optional)
31:22	RO	0	RESERVED

### B.6.1.3 UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RWS	0	DATA LINK PROTOCOL ERROR MASK
5	RO	0	SURPRISE DOWN ERROR MASK (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP MASK
13	RO	0	FLOW CONTROL PROTOCOL ERROR MASK (optional)
14	RWS	0	COMPLETION TIMEOUT MASK
15	RO	0	COMPLETER ABORT MASK (optional)
16	RWS	0	UNEXPECTED COMPLETION MASK
17	RO	0	RECEIVER OVERFLOW MASK (optional)
18	RWS	0	MALFORMED TLP MASK
19	RO	0	ECRC ERROR MASK (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR MASK
21	RO	0	ACS VIOLATION MASK (optional)
31:22	RO	0	RESERVED

### B.6.1.4 UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RWS	1	DATA LINK PROTOCOL ERROR SEVERITY
5	RO	1	SURPRISE DOWN ERROR SEVERITY (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP SEVERITY
13	RO	1	FLOW CONTROL PROTOCOL ERROR SEVERITY (optional)
14	RWS	0	COMPLETION TIMEOUT ERROR SEVERITY
15	RO	0	COMPLETER ABORT ERROR SEVERITY (optional)
16	RWS	0	UNEXPECTED COMPLETION ERROR SEVERITY
17	RO	1	RECEIVER OVERFLOW ERROR SEVERITY (optional)
18	RWS	1	MALFORMED TLP SEVERITY
19	RO	0	ECRC ERROR SEVERITY (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR SEVERITY
21	RO	0	ACS VIOLATION SEVERITY (optional)
31:22	RO	0	RESERVED

### B.6.1.5 CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR STATUS (optional)
5:1	RO	0	RESERVED
6	RW1CS	0	BAD TLP STATUS
7	RW1CS	0	BAD DLLP STATUS
8	RW1CS	0	REPLAY_NUM ROLLOVER STATUS
11:9	RO	0	RESERVED
12	RW1CS	0	REPLAY TIMER TIMEOUT STATUS
13	RW1CS	0	ADVISORY NON-FATAL ERROR STATUS
31:14	RO	0	RESERVED

### B.6.1.6 CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR MASK (optional)
5:1	RO	0	RESERVED
6	RWS	0	BAD TLP MASK
7	RWS	0	BAD DLLP MASK
8	RWS	0	REPLAY_NUM ROLLOVER MASK
11:9	RO	0	RESERVED
12	RWS	0	REPLAY TIMER TIMEOUT MASK
13	RWS	1	ADVISORY NON-FATAL ERROR MASK
31:14	RO	0	RESERVED

### B.6.1.7 ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
4:0	ROS	0	FIRST ERROR POINTER
5	RO	0	ECRC GENERATION CAPABLE
6	RO	0	ECRC GENERATION ENABLE
7	RO	0	ECRC CHECK CAPABLE
8	RWS	0	ECRC CHECK ENABLE
31:9	RO	0	RESERVED

### B.6.1.8 HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
127:0	ROS	0	Header of TLP For First Error

## B.7. PCI Express Virtual Channel Capability

### B.7.1. Standard Endpoint PCI Express Virtual Channel Capability

#### B.7.1.1 PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0002h	VC EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	600h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)

#### B.7.1.2 PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	111b	Extended VC Count (8 VCs)
3	RO	0	Reserved
6:4	RO	100b	Low Priority Extended VC Count
7	RO	0	Reserved
9:8	RO	0	Reference Clock (Reserved For Endpoints)
11:10	RO	0	Port Arbitration Table Entry Size (Reserved For Endpoints)
31:12	RO	0	Reserved

#### B.7.1.3 PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00000010b	VC Arbitration Capability
23:8	RO	0	Reserved
31:24	RO	70h	VC Arbitration Table Offset

**B.7.1.4 PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Load VC Arbitration Table (always returns 0)
3:1	RO	001b	VC Arbitration Select
15:4	RO	0	Reserved

**B.7.1.5 PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	VC Arbitration Table Status
15:1	RO	0	Reserved

**B.7.1.6 VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### B.7.1.7 VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	TC/VC Map (0)
7:1	RO	1111111b	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RO	0	VC ID
30:27	RO	0	Reserved
31	RO	1	VC Enable

### B.7.1.8 Reserved Register (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### B.7.1.9 VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### B.7.1.10 VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### B.7.1.11 VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	1	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### B.7.1.12 Reserved Register (OFFSET 24h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved



**B.7.1.13 VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.14 VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

**B.7.1.15 VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	2	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

**B.7.1.16 Reserved Register (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	Reserved

**B.7.1.17 VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.18 VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### B.7.1.19 VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	3	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### B.7.1.20 Reserved Register (OFFSET 3Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### B.7.1.21 VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### B.7.1.22 VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### B.7.1.23 VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	4	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### B.7.1.24 Reserved Register (OFFSET 48h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

**B.7.1.25 VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.26 VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

**B.7.1.27 VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	5	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

**B.7.1.28 Reserved Register (OFFSET 54h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	Reserved

**B.7.1.29 VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.30 VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### B.7.1.31 VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	6	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### B.7.1.32 Reserved Register (OFFSET 60h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### B.7.1.33 VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.34 VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

**B.7.1.35 VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	7	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

**B.7.1.36 Reserved Register (OFFSET 6Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	Reserved



**B.7.1.37 VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**B.7.1.38 VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
2:0	RW	0	VC ID Phase 0
3	RW	0	Reserved

...

126:124	RW	0	VC ID Phase 31
127	RW	0	Reserved

**B.8. PCI Express Power Budgeting Capability****B.8.1. Standard Device PCI Express Power Budgeting Capability****B.8.1.1 PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0004h	Power Budgeting Extended Capability ID
19:16	RO	1h	Capability Version
31:20	RO	000h	Next Capability Offset (Last capability in list)

**B.8.1.2 DATA SELECT REGISTER (OFFSET 04h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Data Select
31:8	RO	0	Reserved

**B.8.1.3 DATA REGISTER (OFFSET 08h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h*	Base Power (*value depends on Data Select value)
9:8	RO	00b	Data Scale
12:10	RO	000b	PM Sub State
14:13	RO	00b	PM State
17:15	RO	000b*	Type (*value depends on Data Select value)
20:18	RO	000b	Power Rail
31:21	RO	0	Reserved

**B.8.1.4 POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	System Allocated
31:1	RO	0	Reserved

The default power budgeting capability contains the following Data Select Index and Data pairs.

Data Select Index: 0  
 Base Power 25  
 Data Scale 00b (1.0)  
 PM Sub State 000b (Default)  
 PM State 00b (D0)  
 Type 111b (Max)  
 Power Rail 000b (12 Volts)

Data Select Index: 1  
 Base Power 20  
 Data Scale 00b (1.0)  
 PM Sub State 000b (Default)  
 PM State 00b (D0)  
 Type 011b (Sustained)  
 Power Rail 000b (12 Volts)

## Appendix D: PCI Express 3.0 Register and Capability – Defaults, Contents, and Characteristics for Functions in Test Cases

### C.1. Type 0 Configuration Header

#### C.1.1. Standard Upstream Port Type 0 Configuration Header

##### C.1.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance

##### C.1.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

### C.1.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6	RO	0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	00b	DEVSEL TIMING
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

### C.1.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	00000b	MINOR REVISION ID
7:5	RO	001b	MAJOR REVISION ID

**C.1.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PROG I/F CODE
15:8	RO	00h	SUB-CLASS CODE
23:16	RO	FFh	CLASS CODE

**C.1.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	CACHE LINE SIZE

**C.1.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	LATENCY TIMER

**C.1.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
6:0	RO	0000000b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

**C.1.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED

### C.1.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 24h)

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

#### C.1.1.10.1. Default IO BAR - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

#### C.1.1.10.2. Default 32 bit Memory BAR - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
3	RO	0	Prefetchable.
31:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

#### C.1.1.10.3. Default 64 bit Memory BAR – 2 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

### C.1.1.11 PCI CARDBUS CIS POINTER REGISTER (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	CARDBUS INFORMATION STRUCTURE POINTER

### C.1.1.12 PCI SUBSYSTEM VENDOR ID REGISTER (OFFSET 2Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0000h	SUBSYSTEM VENDOR ID

### C.1.1.13 PCI SUBSYSTEM ID REGISTER (OFFSET 2Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0000h	SUBSYSTEM ID

### C.1.1.14 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 30h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

### C.1.1.15 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

### C.1.1.16 PCI RESERVED REGISTER (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	RESERVED

### C.1.1.17 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Interrupt Line

### C.1.1.18 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	Interrupt Pin (INTA supported)

### C.1.1.19 PCI MIN\_GNT/MAX\_LAT REGISTERS (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED

## C.2. Type 1 Configuration Header

### C.2.1. Standard Bridge Device Type 1 Configuration Header

#### C.2.1.1 PCI DEVICE ID-VENDOR ID REGISTER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	XXXXh	VENDOR ID Specified for each specific instance
31:16	RO	XXXXh	DEVICE ID Specified for each specific instance



### C.2.1.2 PCI COMMAND REGISTER (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	I/O SPACE ENABLE
1	RW	0	MEMORY SPACE ENABLE
2	RW	0	BUS MASTER ENABLE
3	RO	0	SPECIAL CYCLE ENABLE
4	RO	0	MEMORY WRITE AND INVALIDATE
5	RO	0	VGA PALETTE SNOOP
6	RW	0	PARITY ERROR RESPONSE
7	RO	0	IDSEL STEPPING/WAIT CYCLE CONTROL
8	RW	0	SERR# ENABLE
9	RO	0	FAST BACK-TO-BACK TRANSACTIONS ENABLE
10	RW	0	INTERRUPT DISABLE
15:11	RO	0	RESERVED

### C.2.1.3 PCI STATUS REGISTER (OFFSET 06h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	0	RESERVED
3	RO	0	INTERRUPT STATUS
4	RO	1	CAPABILITIES LIST
5	RO	0	66MHZ CAPABLE
6	RO	0	RESERVED
7	RO	0	FAST BACK TO BACK CAPABLE
8	RW1C	0	MASTER DATA PARITY ERROR
10:9	RO	00b	DEVSEL TIMING
11	RW1C	0	SIGNALED TARGET ABORT
12	RW1C	0	RECEIVED TARGET ABORT
13	RW1C	0	RECEIVED MASTER ABORT
14	RW1C	0	SIGNALED SYSTEM ERROR
15	RW1C	0	DETECTED PARITY ERROR

**C.2.1.4 PCI REVISION ID REGISTER (OFFSET 08h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	00000b	MINOR REVISION ID
7:5	RO	001b	MAJOR REVISION ID

**C.2.1.5 PCI CLASS CODE REGISTER (OFFSET 09h) – 3 BYTES**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PROG I/F CODE
15:8	RO	04h	SUB-CLASS CODE
23:16	RO	06h	CLASS CODE

**C.2.1.6 PCI CACHE LINE SIZE REGISTER (OFFSET 0Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	CACHE LINE SIZE

**C.2.1.7 PCI MASTER LATENCY TIMER REGISTER (OFFSET 0Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	PRIMARY LATENCY TIMER

**C.2.1.8 PCI HEADER TYPE REGISTER (OFFSET 0Eh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
6:0	RO	0000001b	HEADER TYPE
7	RO	0	MULTI-FUNCTION

### C.2.1.9 PCI BIST REGISTER (OFFSET 0Fh) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	BIST STATUS
5:4	RO	0	RESERVED
6	RO	0	INVOKE BIST
7	RO	0	BIST SUPPORTED

### C.2.1.10 PCI BASE ADDRESS REGISTERS (OFFSET 10h – 18h)

BAR registers are specified individually for each device in a test topology. The default register contents for BARs of various types are shown here for reference.

#### C.2.1.10.1. Default IO BAR - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	I/O Space Indicator
1	RO	0	Reserved
31:2	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

#### C.2.1.10.2. Default 32 bit Memory BAR - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	00b	Type (Locate anywhere in 32 bit address space)
3	RO	0	Prefetchable.
31:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

#### C.2.1.10.3. Default 64 bit Memory BAR – 2 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Memory Space Indicator
2:1	RO	10b	Type (Locate anywhere in 64 bit address space)
3	RO	0	Prefetchable.
63:4	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

**C.2.1.11 PCI PRIMARY BUS NUMBER REGISTER (OFFSET 18h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Primary Bus Number

**C.2.1.12 PCI SECONDARY BUS NUMBER REGISTER (OFFSET 19h) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Secondary Bus Number

**C.2.1.13 PCI SUBORDINATE BUS NUMBER REGISTER (OFFSET 1Ah) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RW	00h	Subordinate Bus Number

**C.2.1.14 PCI SECONDARY LATENCY TIMER REGISTER (OFFSET 1Bh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h	Secondary Latency Timer

**C.2.1.15 PCI I/O BASE REGISTER****C.2.1.15.1. I/O Base Register 16 bit Address Support (OFFSET 1Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Base Address 15:12

**C.2.1.15.2. I/O Base Register 32 bit Address Support (OFFSET 1Ch) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Base Address 15:12

**C.2.1.16 PCI I/O Limit REGISTER****C.2.1.16.1. I/O Limit Register 16 bit Address Support (OFFSET 1Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0h	I/O Addressing Capability (16 bit)
7:4	RW	0h	I/O Limit Address 15:12

**C.2.1.16.2. I/O Limit Register 32 bit Address Support (OFFSET 1Dh) - BYTE**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	1h	I/O Addressing Capability (32 bit)
7:4	RW	0h	I/O Limit Address 15:12

**C.2.1.17 PCI SECONDARY STATUS REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
4:0	RO	0	Reserved
5	RO	0	66 MHz Capable
6	RO	0	Reserved
7	RO	0	Fast Back-to-Back Transactions Capable
8	RW1C	0	Master Data Parity Error
10:9	RO	00b	DEVSEL Timing
11	RW1C	0	Signaled Target Abort
12	RW1C	0	Received Target Abort
13	RW1C	0	Received Master Abort
14	RW1C	0	Received System Error
15	RW1C	0	Detected Parity Error

**C.2.1.18 PCI MEMORY BASE REGISTER (OFFSET 20h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0	Reserved
15:4	RW	0	Memory Base Address Bits 31:20

**C.2.1.19 PCI MEMORY LIMIT REGISTER (OFFSET 22h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0	Reserved
15:4	RW	0	Memory Limit Address Bits 31:20

**C.2.1.20 PCI PREFETCHABLE MEMORY BASE****C.2.1.20.1. PREFETCHABLE MEMORY BASE REGISTER 32 Bit Address Support (OFFSET 24h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

**C.2.1.20.2. PREFETCHABLE MEMORY BASE REGISTER 64 Bit Address Support (OFFSET 24h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Base Address Bits 31:20

## C.2.1.21 PCI PREFETCHABLE MEMORY LIMIT

### C.2.1.21.1. PREFETCHABLE MEMORY LIMIT REGISTER 32 Bit Address Support (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Address Space Support (32 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

### C.2.1.21.2. PREFETCHABLE MEMORY LIMIT REGISTER 64 Bit Address Support (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0001b	Address Space Support (64 bit)
15:4	RW	0	Prefetchable Memory Limit Address Bits 31:20

## C.2.1.22 PCI PREFETCHABLE MEMORY BASE UPPER 32

### C.2.1.22.1. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 32 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Base Address Upper 32 Bits

### C.2.1.22.2. PREFETCHABLE MEMORY BASE UPPER 32 REGISTER 64 Bit Address Support (OFFSET 28h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Base Address Upper 32 Bits

## C.2.1.23 PCI PREFETCHABLE MEMORY LIMIT UPPER 32

### C.2.1.23.1. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 32 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Prefetchable Memory Limit Address Upper 32 Bits

### C.2.1.23.2. PREFETCHABLE MEMORY LIMIT UPPER 32 REGISTER 64 Bit Address Support (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	Prefetchable Memory Limit Address Upper 32 Bits

## C.2.1.24 PCI I/O BASE UPPER 16

### C.2.1.24.1. I/O BASE UPPER 16 REGISTER 16 Bit Address Support (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	I/O Base Address Upper 16 Bits

### C.2.1.24.2. I/O BASE UPPER 16 REGISTER 32 Bit Address Support (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0	I/O Base Address Upper 16 Bits

## C.2.1.25 PCI I/O LIMIT UPPER 16

### C.2.1.25.1. I/O LIMIT UPPER 16 REGISTER 16 Bit Address Support (OFFSET 32h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	I/O Limit Address Upper 16 Bits

### C.2.1.25.2. I/O LIMIT UPPER 16 REGISTER 32 Bit Address Support (OFFSET 32h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0	I/O Limit Address Upper 16 Bits



### C.2.1.26 PCI CAPABILITY POINTER REGISTER (OFFSET 34h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	40h	CAPABILITIES POINTER (Pointer to Power Management Capability)
31:8	RO	0	RESERVED

### C.2.1.27 PCI EXPANSION ROM BASE ADDRESS REGISTER (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Expansion ROM Access Enable
10:1	RO	0	Reserved
31:11	RO*	0	Default value is zero. By default all bits are RO. (*All bits used in determining size are RW)

### C.2.1.28 PCI INTERRUPT LINE REGISTER (OFFSET 3Ch) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Interrupt Line

### C.2.1.29 PCI INTERRUPT PIN REGISTER (OFFSET 3Dh) – BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	Interrupt Pin (INTA supported)

### C.2.1.30 PCI BRIDGE CONTROL REGISTER (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Parity Error Response Enable
1	RW	0	SERR# Enable
2	RW	0	ISA Enable
3	RW	0	VGA Enable
4	RW	0	VGA 16-bit decode
5	RO	0	Master Abort Mode
6	RW	0	Secondary Bus Reset
7	RO	0	Fast Back-to-Back Transactions Enable
8	RO	0	Primary Discard Timer
9	RO	0	Secondary Discard Timer
10	RO	0	Discard Timer Status
11	RO	0	Discard Timer SERR# Enable
15:12	RO	0	Reserved

## C.3. Power Management Capability

### C.3.1. Standard Type 0 Header Device Power Management Capability Structure

#### C.3.1.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### C.3.1.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	011b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

### C.3.1.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
2	RO	0	RESERVED
3	RO	0	NO SOFT RESET
7:4	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### C.3.1.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### C.3.1.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## C.3.2. Standard Type 1 Header Device Power Management Capability Structure

### C.3.2.1 PM CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	01h	PM CAPABILITY ID
15:8	RO	50h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### C.3.2.2 POWER MANAGEMENT CAPABILITIES (PMC) REGISTER (OFFSET 02h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	011b	VERSION
3	RO	0	PME CLOCK
4	RO	0	RESERVED
5	RO	0	DEVICE SPECIFIC INITIALIZATION
8:6	RO	000b	AUX CURRENT
9	RO	0	D1 SUPPORT
10	RO	0	D2 SUPPORT
15:11	RO	11001b	PME SUPPORT

### C.3.2.3 POWER MANAGEMENT CONTROL/STATUS REGISTER (PMCSR) (OFFSET 04h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	POWER STATE
2	RO	0	RESERVED
3	RO	0	NO SOFT RESET
7:4	RO	0	RESERVED
8	RWS	0	PME ENABLE
12:9	RO	0000b	DATA SELECT
14:13	RO	00b	DATA SCALE
15	RW1C	0	PME STATUS (Set if a PME is asserted by the device)

### C.3.2.4 PM BRIDGE SUPPORT EXTENSIONS REGISTER (PMCSR\_BSE) (OFFSET 06h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
5:0	RO	0	RESERVED
6	RO	0	B2_B3#
7	RO	0	BPCC_EN

### C.3.2.5 POWER MANAGEMENT DATA REGISTER (OFFSET 07h) - BYTE

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00h	DATA

## C.4. MSI Capability Structures

### C.4.1. Standard Device PCI MSI-32 CAPABILITY STRUCTURE

#### C.4.1.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### C.4.1.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	0	64-BIT ADDRESS CAPABLE
8	RO	0	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

#### C.4.1.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	32-BIT ADDRESS

#### C.4.1.4 MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

## C.4.2. Standard Device PCI MSI-64 CAPABILITY STRUCTURE

### C.4.2.1 MSI CAPABILITY FIELDS (OFFSET 00h)

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

### C.4.2.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	1	64-BIT ADDRESS CAPABLE
8	RO	0	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

### C.4.2.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	LOWER 32-BIT ADDRESS

### C.4.2.4 MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	UPPER 32-BIT ADDRESS

**C.4.2.5 MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RW	0000h	MESSAGE DATA

**C.4.3. Standard Device PCI MSI-PVM-32 CAPABILITY STRUCTURE****C.4.3.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

**C.4.3.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	0	64-BIT ADDRESS CAPABLE
8	RO	1	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

**C.4.3.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0:1	RO	0	RESERVED
31:2	RW	0	32-BIT ADDRESS



**C.4.3.4 MSI MESSAGE DATA REGISTER (OFFSET 08h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RW	0000h	MESSAGE DATA

**C.4.3.5 MSI RESERVED REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.4.3.6 MSI MASK BITS REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RW	0	MASK BITS

**C.4.3.7 MSI PENDING BITS REGISTER (OFFSET 10h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	PENDING BITS

**C.4.4. Standard Device PCI MSI-PVM-64 CAPABILITY STRUCTURE****C.4.4.1 MSI CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	05h	MSI CAPABILITY ID
15:8	RW	60h	NEXT CAPABILITY POINTER (Points to the next capability implemented)

#### C.4.4.2 MSI MESSAGE CONTROL REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	MSI ENABLE
3:1	RO	000b	MULTIPLE MESSAGE CAPABLE
6:4	RO	000b	MULTIPLE MESSAGE ENABLE
7	RO	1	64-BIT ADDRESS CAPABLE
8	RO	1	PER-VECTOR MASKING CAPABLE
15:9	RO	0	RESERVED

#### C.4.4.3 MSI MESSAGE ADDRESS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0:1	RO	0	RESERVED
31:2	RW	0	LOWER 32-BIT ADDRESS

#### C.4.4.4 MSI MESSAGE Upper ADDRESS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	UPPER 32-BIT ADDRESS

#### C.4.4.5 MSI MESSAGE DATA REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RW	0000h	MESSAGE DATA

#### C.4.4.6 MSI RESERVED REGISTER (OFFSET 0Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED

#### C.4.4.7 MSI MASK BITS REGISTER (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RW	0	MASK BITS

**C.4.4.8 MSI PENDING BITS REGISTER (OFFSET 14h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	PENDING BITS

**C.5. PCI Express Capability Structure****C.5.1. Standard Upstream Port Device PCI Express Capability Structure****C.5.1.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

**C.5.1.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	2h	CAPABILITY VERSION
7:4	RO	0001b	DEVICE/PORT TYPE (Legacy PCI Express Endpoint. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED
13:9	RO	0	INTERRUPT MESSAGE NUMBER
14	RO	0	UNDEFINED
15	RO	0	RESERVED

### C.5.1.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	CAPTURED SLOT POWER LIMIT VALUE
27:26	RO	0	CAPTURED SLOT POWER LIMIT SCALE
28	RO	0	FUNCTION LEVEL RESET CAPABILITY
31:29	RO	0	RESERVED

### C.5.1.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	INITIATE FUNCTION LEVEL RESET (always returns 0)

**C.5.1.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Set to 1 when non-posted requests are not yet completed and clear when they are completed)
15:6	RO	0	RESERVED

**C.5.1.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RO	0011b	MAX LINK SPEED
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	0	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE
21	RO	0	LINK BANDWIDTH NOTIFICATION CAPABILITY
22	RO	1	ASPM OPTIONALITY COMPLIANCE
23	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER

### C.5.1.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RO	0	LINK DISABLE
5	RO	0	RETRAIN LINK
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
9	RO	0	HARDWARE AUTONOMOUS WIDTH DISABLE
10	RO	0	LINK BANDWIDTH MANAGEMENT INTERRUPT ENABLE
11	RO	0	LINK AUTONOMOUS BANDWIDTH INTERRUPT ENABLE
15:12	RO	0	RESERVED

### C.5.1.8 LINK STATUS REGISTER (OFFSET 12h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0011b	CURRENT LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	0	DATA LINK LAYER LINK ACTIVE
14	RO	0	LINK BANDWIDTH MANAGEMENT STATUS
15	RO	0	LINK AUTONOMOUS BANDWIDTH STATUS

### C.5.1.9 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	RESERVED

**C.5.1.10 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.1.11 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.1.12 ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.1.13 ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.1.14 ROOT STATUS REGISTER (OFFSET 20h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

### C.5.1.15 DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Ranges Supported
4	RO	0	Completion Timeout Disable Supported
5	RO	0	ARI Forwarding Supported
6	RO	0	AtomicOp Routing Supported
7	RO	0	32-bit AtomicOp Completer Supported
8	RO	0	64-bit AtomicOp Completer Supported
9	RO	0	128-bit CAS Completer Supported
10	RO	0	No RO-enabled PR-PR Passing
11	RO	0	LTR Mechanism Supported
13:12	RO	00b	TPH Completer Supported
17:14	RO	0	RESERVED
19:18	RO	00b	OBFF Supported
20	RO	0	Extended Fmt Field Supported
21	RO	0	End-End TLP Prefix Supported
23:22	RO	00b	Max End-End TLP Prefixes
31:24	RO	0	RESERVED

### C.5.1.16 DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Value
4	RO	0	Completion Timeout Disable
5	RO	0	ARI Forwarding Enable
6	RO	0	AtomicOp Request Enable
7	RO	0	AtomicOp Egress Blocking
8	RO	0	IDO Request Enable
9	RO	0	IDO Completion Enable
10	RO	0	LTR Mechanism Enable
12:11	RO	0	RESERVED
14:13	RO	00b	OBFF Enable
15	RW	0	End-End TLP Prefix Blocking

### C.5.1.17 DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	RESERVED



### C.5.1.18 LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RESERVED
7:1	RO	0000111b	Supported Link Speeds Vector
8	RO	0	Crosslink Supported
31:9	RO	0	RESERVED

### C.5.1.19 LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RWS*	0011b*	Target Link Speed (*Must be RO-Zero for non-zero function number)
4	RWS*	0	Enter Compliance (*Must be RO-Zero for non-zero function number)
5	RO	0	Hardware Autonomous Speed Disable
6	RO	0	Selectable De-emphasis
9:7	RWS*	000b	Transmit Margin (*Must be RO-Zero for non-zero function number)
10	RWS*	0	Enter Modified Compliance (*Must be RO-Zero for non-zero function number)
11	RWS*	0	Compliance SOS (*Must be RO-Zero for non-zero function number)
15:12	RWS*	0000b	Compliance Preset/De-emphasis (*Must be RO-Zero for non-zero function number)

**C.5.1.20 LINK STATUS 2 REGISTER (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	Current De-emphasis Level (test card is -3.5dB)
1	ROS	0	Equalization Complete
2	ROS	0	Equalization Phase 1 Successful
3	ROS	0	Equalization Phase 2 Successful
4	ROS	0	Equalization Phase 3 Successful
5	RW1CS*	0	Link Equalization Request (*Must be RO-Zero for non-zero function number)
15:6	RO	0	RESERVED

**C.5.1.21 SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**C.5.1.22 SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.1.23 SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

## C.5.2. Standard Downstream Port Device PCI Express Capability Structure

### C.5.2.1 PCI EXPRESS CAPABILITY ID-NEXT CAPABILITY POINTER (OFFSET 00h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	10h	PCI EXPRESS CAPABILITY ID
15:8	RO	00h	NEXT CAPABILITY POINTER (Last capability in list)

### C.5.2.2 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 02h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	2h	CAPABILITY VERSION
7:4	RO	0110b	DEVICE/PORT TYPE (Downstream Port of PCI Express Switch. Other types will be noted in individual test cases.)
8	RO	0	SLOT IMPLEMENTED (Other value will be noted in individual test cases)
13:9	RO	0	INTERRUPT MESSAGE NUMBER
14	RO	0	UNDEFINED
15	RO	0	RESERVED

### C.5.2.3 DEVICE CAPABILITIES REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	000b	MAX_PAYLOAD_SIZE SUPPORTED
4:3	RO	00b	PHANTOM FUNCTIONS SUPPORTED
5	RO	0	EXTENDED TAG FIELD SUPPORTED
8:6	RO	000b	ENDPOINT L0s ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
11:9	RO	000b	ENDPOINT L1 ACCEPTABLE LATENCY (must be 0 for non-Endpoints)
12	RO	0	UNDEFINED
13	RO	0	UNDEFINED
14	RO	0	UNDEFINED
15	RO	1	ROLE-BASED ERROR REPORTING
17:16	RO	0	RESERVED
25:18	RO	0	RESERVED (Captured Slot Power Limit Value for Upstream Ports)
27:26	RO	0	RESERVED (Captured Slot Power Limit Scale for Upstream Ports)
31:28	RO	0	RESERVED

### C.5.2.4 DEVICE CONTROL REGISTER (OFFSET 08h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	CORRECTABLE ERROR REPORTING ENABLE
1	RW	0	NON-FATAL ERROR REPORTING ENABLE
2	RW	0	FATAL ERROR REPORTING ENABLE
3	RW	0	UNSUPPORTED REQUEST REPORTING ENABLE
4	RW	1	ENABLE RELAXED ORDERING
7:5	RO	000b	MAX_PAYLOAD_SIZE
8	RO	0	EXTENDED TAG FIELD ENABLE
9	RO	0	PHANTOM FUNCTIONS ENABLE
10	RO	0	AUXILIARY (AUX) POWER PM ENABLE
11	RW	1	ENABLE NO SNOOP
14:12	RW	010b	MAX_READ_REQUEST_SIZE
15	RO	0	RESERVED

### C.5.2.5 DEVICE STATUS REGISTER (OFFSET 0Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1C	0	CORRECTABLE ERROR DETECTED
1	RW1C	0	NON-FATAL ERROR DETECTED
2	RW1C	0	FATAL ERROR DETECTED
3	RW1C	0	UNSUPPORTED REQUEST DETECTED
4	RO	0	AUX POWER DETECTED
5	RO	0	TRANSACTIONS PENDING (Not used by default downstream device)
15:6	RO	0	RESERVED

### C.5.2.6 LINK CAPABILITIES REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0011b	MAX LINK SPEED
9:4	RO	000001b	MAXIMUM LINK WIDTH (x1 is the width of the test card)
11:10	RO	11b	ACTIVE STATE POWER MANAGEMENT (ASPM) SUPPORT
14:12	RO	111b	L0s EXIT LATENCY
17:15	RO	111b	L1 EXIT LATENCY
18	RO	0	CLOCK POWER MANAGEMENT
19	RO	0	SURPRISE DOWN ERROR REPORTING CAPABLE
20	RO	1	DATA LINK LAYER LINK ACTIVE REPORTING CAPABLE (If Downstream port supports 8.0 GT/s this bit must be 1)
21	RO	1	LINK BANDWIDTH NOTIFICATION CAPABILITY (must be 1 for 8.0 GT/s or 5.0 GT/s capable downstream port)
22	RO	1	ASPM OPTIONALITY COMPLIANCE
23	RO	0	RESERVED
31:24	RO	00h	PORT NUMBER

### C.5.2.7 LINK CONTROL REGISTER (OFFSET 10h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
1:0	RW	00b	ACTIVE STATE POWER MANAGEMENT (ASPM) CONTROL
2	RO	0	RESERVED
3	RO	0	READ COMPLETION BOUNDARY (RCB)
4	RW	0	LINK DISABLE
5	RO	0	RETRAIN LINK (always returns 0)
6	RW	0	COMMON CLOCK CONFIGURATION
7	RW	0	EXTENDED SYNCH
8	RO	0	ENABLE CLOCK POWER MANAGEMENT
9	RO	0	HARDWARE AUTONOMOUS WIDTH DISABLE
10	RW	0	LINK BANDWIDTH MANAGEMENT INTERRUPT ENABLE
11	RW	0	LINK AUTONOMOUS BANDWIDTH INTERRUPT ENABLE
15:12	RO	0	RESERVED

### C.5.2.8 LINK STATUS REGISTER (OFFSET 12h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0011b	CURRENT LINK SPEED
9:4	RO	000001b	NEGOTIATED LINK WIDTH (x1 is the width of the test card)
10	RO	0	UNDEFINED
11	RO	0	LINK TRAINING
12	RO	1	SLOT CLOCK CONFIGURATION
13	RO	1	DATA LINK LAYER LINK ACTIVE (If Downstream port supports 8.0 GT/s this bit must be supported; DL_Active state = 1)
14	RW1C	0	LINK BANDWIDTH MANAGEMENT STATUS
15	RW1C	0	LINK AUTONOMOUS BANDWIDTH STATUS

### C.5.2.9 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	RESERVED

**C.5.2.10 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.2.11 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
5:0	RO	0	RESERVED
6	RO	1	Presence Detect State (If no slot present must be 1)
15:7	RO	0	RESERVED

**C.5.2.12 ROOT CONTROL REGISTER (OFFSET 1Ch) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.2.13 ROOT CAPABILITIES REGISTER (OFFSET 1Eh) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.2.14 ROOT STATUS REGISTER (OFFSET 20h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

### C.5.2.15 DEVICE CAPABILITIES 2 REGISTER (OFFSET 24h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Ranges Supported
4	RO	0	Completion Timeout Disable Supported
5	RO	0	ARI Forwarding Supported
6	RO	0	AtomicOp Routing Supported
7	RO	0	32-bit AtomicOp Completer Supported
8	RO	0	64-bit AtomicOp Completer Supported
9	RO	0	128-bit CAS Completer Supported
10	RO	0	No RO-enabled PR-PR Passing
11	RO	0	LTR Mechanism Supported
13:12	RO	00b	TPH Completer Supported
17:14	RO	0	RESERVED
19:18	RO	00b	OBFF Supported
20	RO	0	Extended Fmt Field Supported
21	RO	0	End-End TLP Prefix Supported
23:22	RO	00b	Max End-End TLP Prefixes
31:24	RO	0	RESERVED

### C.5.2.16 DEVICE CONTROL 2 REGISTER (OFFSET 28h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Completion Timeout Value
4	RO	0	Completion Timeout Disable
5	RO	0	ARI Forwarding Enable
6	RO	0	AtomicOp Request Enable
7	RO	0	AtomicOp Egress Blocking
8	RO	0	IDO Request Enable
9	RO	0	IDO Completion Enable
10	RO	0	LTR Mechanism Enable
12:11	RO	0	RESERVED
14:13	RO	00b	OBFF Enable
15	RW	0	End-End TLP Prefix Blocking



**C.5.2.17 DEVICE STATUS 2 REGISTER (OFFSET 2Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.2.18 LINK CAPABILITIES 2 REGISTER (OFFSET 2Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	RESERVED
7:1	RO	0000111b	Supported Link Speeds Vector
8	RO	0	Crosslink Supported
31:9	RO	0	RESERVED

**C.5.2.19 LINK CONTROL 2 REGISTER (OFFSET 30h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
3:0	RWS	0011b	Target Link Speed
4	RWS	0	Enter Compliance
5	RO	0	Hardware Autonomous Speed Disable
6	RO	1	Selectable De-emphasis (test card is -3.5dB)
9:7	RWS	000b	Transmit Margin
10	RWS	0	Enter Modified Compliance
11	RWS	0	Compliance SOS
15:12	RWS	0000b	Compliance Preset/De-emphasis

**C.5.2.20 LINK STATUS 2 REGISTER (OFFSET 32h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	1	Current De-emphasis Level (test card is -3.5dB)
1	ROS	0	Equalization Complete
2	ROS	0	Equalization Phase 1 Successful
3	ROS	0	Equalization Phase 2 Successful
4	ROS	0	Equalization Phase 3 Successful
5	RW1CS	0	Link Equalization Request
15:6	RO	0	RESERVED

**C.5.2.21 SLOT CAPABILITIES 2 REGISTER (OFFSET 34h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
31:0	RO	0	RESERVED

**C.5.2.22 SLOT CONTROL 2 REGISTER (OFFSET 38h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

**C.5.2.23 SLOT STATUS 2 REGISTER (OFFSET 3Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	RESERVED

### C.5.3. Standard Downstream Port Device with Slot PCI Express Capability Structure

The registers for a default downstream port device with a slot are identical to those for a default downstream device without a slot from offset 00h through 3Bh inclusive, with the following exceptions:

#### C.5.3.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
8	RO	1	SLOT IMPLEMENTED

#### C.5.3.2 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Attention Button Present
1	RO	0	Power Controller Present
2	RO	0	MRL Sensor Present
3	RO	0	Attention Indicator Present
4	RO	0	Power Indicator Present
5	RO	0	Hot-Plug Surprise
6	RO	0	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale
17	RO	0	Electromechanical Interlock Present
18	RO	0	No Command Completed Support
31:19	RO	0	Physical Slot Number

**C.5.3.3 SLOT CONTROL REGISTER (OFFSET 18h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Attention Button Pressed Enable
1	RO	0	Power Fault Detected Enable
2	RO	0	MRL Sensor Changed Enable
3	RO	0	Presence Detect Changed Enable
4	RO	0	Command Completed Interrupt Enable
5	RO	0	Hot-Plug Interrupt Enable
7:6	RW	00b	Attention Indicator Control
9:8	RW	00b	Power Indicator Control
10	RO	0	Power Controller Control (No Power Controller Implemented By Default)
11	RO	0	Electromechanical Interlock Control
12	RW	0	Data Link Layer State Changed Enable (If Downstream port supports 8.0 GT/s this bit must be supported)
15:13	RO	0	RESERVED

**C.5.3.4 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RW1C	0	Command Completed
5	RO	0	MRL Sensor State
6	RO	1	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status
8	RW1C	0	Data Link Layer State Changed (If Downstream port supports 8.0 GT/s this bit must be supported)
15:9	RO	0	RESERVED

## C.5.4. Standard Downstream Port Device with Hot Plug Capable Slot PCI Express Capability Structure

The registers for a default downstream port device with a hot plug capable slot are identical to those for a default downstream device without a slot from offset 00h through 3Bh inclusive, with the following exceptions:

### C.5.4.1 PCI EXPRESS CAPABILITIES REGISTER (OFFSET 2h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
8	RO	1	SLOT IMPLEMENTED

### C.5.4.2 SLOT CAPABILITIES REGISTER (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	Attention Button Present
1	RO	1	Power Controller Present
2	RO	1	MRL Sensor Present
3	RO	1	Attention Indicator Present
4	RO	1	Power Indicator Present
5	RO	1	Hot-Plug Surprise
6	RO	1	Hot-Plug Capable
14:7	RO	00011001b	Slot Power Limit Value (25W = Default)
16:15	RO	00b	Slot Power Limit Scale
17	RO	1	Electromechanical Interlock Present
18	RO	1	No Command Completed Support
31:19	RO	0	Physical Slot Number

### C.5.4.3 SLOT CONTROL REGISTER (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Attention Button Pressed Enable
1	RW	0	Power Fault Detected Enable
2	RW	0	MRL Sensor Changed Enable
3	RW	0	Presence Detect Changed Enable
4	RW	0	Command Completed Interrupt Enable
5	RW	0	Hot-Plug Interrupt Enable
7:6	RW	11b	Attention Indicator Control
9:8	RW	01b	Power Indicator Control (Off)
10	RW	0	Power Controller Control (On)
11	RO	0	Electromechanical Interlock Control
12	RW	0	Data Link Layer State Changed Enable (If Downstream port supports 8.0 GT/s this bit must be supported)
15:13	RO	0	RESERVED

### C.5.4.4 SLOT STATUS REGISTER (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1C	0	Attention Button Pressed
1	RW1C	0	Power Fault Detected
2	RW1C	0	MRL Sensor Changed
3	RW1C	0	Presence Detect Changed
4	RO	0	Command Completed
5	RO	1	MRL Sensor State (Open)
6	RO	0	Presence Detect State (0b = Slot Empty; 1b = Card Present)
7	RO	0	Electromechanical Interlock Status (Disengaged)
8	RW1C	0	Data Link Layer State Changed (If Downstream port supports 8.0 GT/s this bit must be supported)
15:9	RO	0	RESERVED

## C.6. PCI Express Advanced Error Reporting Capability

### C.6.1. Standard PCI Express Advanced Error Reporting Capability (without TLP Prefix Support)

#### C.6.1.1 PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0001h	AER EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	138h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)

### C.6.1.2 UNCORRECTABLE ERROR STATUS REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RW1CS	0	DATA LINK PROTOCOL ERROR STATUS
5	RO	0	SURPRISE DOWN ERROR STATUS (optional)
11:6	RO	0	RESERVED
12	RW1CS	0	POISONED TLP STATUS
13	RO	0	FLOW CONTROL PROTOCOL ERROR STATUS (optional)
14	RW1CS	0	COMPLETION TIMEOUT STATUS
15	RO	0	COMPLETER ABORT STATUS (optional)
16	RW1CS	0	UNEXPECTED COMPLETION STATUS
17	RO	0	RECEIVER OVERFLOW STATUS (optional)
18	RW1CS	0	MALFORMED TLP STATUS
19	RO	0	ECRC ERROR STATUS (optional)
20	RW1CS	0	UNSUPPORTED REQUEST ERROR STATUS
21	RO	0	ACS VIOLATION STATUS (optional)
22	RO	0	UNCORRECTABLE INTERNAL ERROR STATUS (optional)
23	RO	0	MC BLOCKED TLP STATUS (optional)
24	RO	0	ATOMICOP EGRESS BLOCKED STATUS (optional)
25	RO	0	TLP PREFIX BLOCKED ERROR STATUS (optional)
31:26	RO	0	RESERVED



### C.6.1.3 UNCORRECTABLE ERROR MASK REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RWS	0	DATA LINK PROTOCOL ERROR MASK
5	RO	0	SURPRISE DOWN ERROR MASK (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP MASK
13	RO	0	FLOW CONTROL PROTOCOL ERROR MASK (optional)
14	RWS	0	COMPLETION TIMEOUT MASK
15	RO	0	COMPLETER ABORT MASK (optional)
16	RWS	0	UNEXPECTED COMPLETION MASK
17	RO	0	RECEIVER OVERFLOW MASK (optional)
18	RWS	0	MALFORMED TLP MASK
19	RO	0	ECRC ERROR MASK (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR MASK
21	RO	0	ACS VIOLATION MASK (optional)
22	RO	0	UNCORRECTABLE INTERNAL ERROR MASK (optional)
23	RO	0	MC BLOCKED TLP MASK (optional)
24	RO	0	ATOMICOP EGRESS BLOCKED MASK (optional)
25	RO	0	TLP PREFIX BLOCKED ERROR MASK (optional)
31:26	RO	0	RESERVED

### C.6.1.4 UNCORRECTABLE ERROR SEVERITY REGISTER (OFFSET 0Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	UNDEFINED
3:1	RO	0	RESERVED
4	RWS	1	DATA LINK PROTOCOL ERROR SEVERITY
5	RO	0	SURPRISE DOWN ERROR SEVERITY (optional)
11:6	RO	0	RESERVED
12	RWS	0	POISONED TLP SEVERITY
13	RO	0	FLOW CONTROL PROTOCOL ERROR SEVERITY (optional)
14	RWS	0	COMPLETION TIMEOUT ERROR SEVERITY
15	RO	0	COMPLETER ABORT ERROR SEVERITY (optional)
16	RWS	0	UNEXPECTED COMPLETION ERROR SEVERITY
17	RO	0	RECEIVER OVERFLOW ERROR SEVERITY (optional)
18	RWS	1	MALFORMED TLP SEVERITY
19	RO	0	ECRC ERROR SEVERITY (optional)
20	RWS	0	UNSUPPORTED REQUEST ERROR SEVERITY
21	RO	0	ACS VIOLATION SEVERITY (optional)
22	RO	0	UNCORRECTABLE INTERNAL ERROR SEVERITY (optional)
23	RO	0	MC BLOCKED TLP SEVERITY (optional)
24	RO	0	ATOMICOP EGRESS BLOCKED SEVERITY (optional)
25	RO	0	TLP PREFIX BLOCKED ERROR SEVERITY (optional)
31:26	RO	0	RESERVED

### C.6.1.5 CORRECTABLE ERROR STATUS REGISTER (OFFSET 10h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR STATUS (optional)
5:1	RO	0	RESERVED
6	RW1CS	0	BAD TLP STATUS
7	RW1CS	0	BAD DLLP STATUS
8	RW1CS	0	REPLAY_NUM ROLLOVER STATUS
11:9	RO	0	RESERVED
12	RW1CS	0	REPLAY TIMER TIMEOUT STATUS
13	RW1CS	0	ADVISORY NON-FATAL ERROR STATUS
14	RO	0	CORRECTED INTERNAL ERROR STATUS (optional)
15	RO	0	HEADER LOG OVERFLOW STATUS (optional)
31:16	RO	0	RESERVED

### C.6.1.6 CORRECTABLE ERROR MASK REGISTER (OFFSET 14h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	RECEIVER ERROR MASK (optional)
5:1	RO	0	RESERVED
6	RWS	0	BAD TLP MASK
7	RWS	0	BAD DLLP MASK
8	RWS	0	REPLAY_NUM ROLLOVER MASK
11:9	RO	0	RESERVED
12	RWS	0	REPLAY TIMER TIMEOUT MASK
13	RWS	1	ADVISORY NON-FATAL ERROR MASK
14	RO	0	CORRECTED INTERNAL ERROR MASK (optional)
15	RO	0	HEADER LOG OVERFLOW MASK (optional)
31:16	RO	0	RESERVED

### C.6.1.7 ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
4:0	ROS	0	FIRST ERROR POINTER
5	RO	0	ECRC GENERATION CAPABLE
6	RO	0	ECRC GENERATION ENABLE
7	RO	0	ECRC CHECK CAPABLE
8	RO	0	ECRC CHECK ENABLE
9	RO	0	MULTIPLE HEADER RECORDING CAPABLE
10	RO	0	MULTIPLE HEADER RECORDING ENABLE
11	RO	0	TLP PREFIX LOG PRESENT
31:12	RO	0	RESERVED

### C.6.1.8 HEADER LOG REGISTER (OFFSET 1Ch) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
127:0	ROS	0	Header of TLP For First Error

## C.6.2. Standard PCI Express Advanced Error Reporting Capability with TLP Prefix Support

The registers for a device with AER that supports TLP Prefix Logging are identical to those for a standard AER device without TLP Prefix Logging from offset 00h through 2Bh inclusive, with the following exceptions:

### C.6.2.1 PCI EXPRESS ADVANCED ERROR REPORTING CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
19:16	RO	2h	CAPABILITY VERSION

### C.6.2.2 ADVANCED ERROR CAPABILITIES and CONTROL REGISTER (OFFSET 18h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
11	ROS	0	TLP PREFIX LOG PRESENT

Also, the following additional registers to the PCI Express Advanced Error Reporting Capability structure are implemented with the indicated characteristics.

### C.6.2.3 ROOT ERROR COMMAND REGISTER (OFFSET 2Ch) – DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Reserved

### C.6.2.4 ROOT ERROR STATUS REGISTER (OFFSET 30h) – DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
31:0	RO	0	Reserved

### C.6.2.5 CORRECTABLE ERROR SOURCE IDENTIFICATION REGISTER (OFFSET 34h) –WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.6.2.6 ERROR SOURCE IDENTIFICATION REGISTER (OFFSET 36h) –WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.6.2.7 TLP PREFIX LOG REGISTER (OFFSET 38h) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
127:0	ROS	0	TLP Prefix Log

## C.7. PCI Express Virtual Channel Capability

### C.7.1. Standard Endpoint PCI Express Virtual Channel Capability

#### C.7.1.1 PCI EXPRESS VIRTUAL CHANNEL ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0002h	VC EXTENDED CAPABILITY ID
19:16	RO	1h	CAPABILITY VERSION
31:20	RO	600h	NEXT CAPABILITY OFFSET (Points to the next capability implemented)

#### C.7.1.2 PORT VC CAPABILITY REGISTER 1 (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RO	111b	Extended VC Count (8 VCs)
3	RO	0	Reserved
6:4	RO	100b	Low Priority Extended VC Count
7	RO	0	Reserved
9:8	RO	0	Reference Clock (Reserved For Endpoints)
11:10	RO	0	Port Arbitration Table Entry Size (Reserved For Endpoints)
31:12	RO	0	Reserved

### C.7.1.3 PORT VC CAPABILITY REGISTER 2 (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	00000010b	VC Arbitration Capability
23:8	RO	0	Reserved
31:24	RO	70h	VC Arbitration Table Offset

### C.7.1.4 PORT VC CONTROL REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Load VC Arbitration Table (always returns 0)
3:1	RO	001b	VC Arbitration Select
15:4	RO	0	Reserved

### C.7.1.5 PORT VC STATUS REGISTER (OFFSET 0Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	VC Arbitration Table Status
15:1	RO	0	Reserved

### C.7.1.6 VC RESOURCE CAPABILITY REGISTER 0 (OFFSET 10h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.7 VC RESOURCE CONTROL REGISTER 0 (OFFSET 14h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	1	TC/VC Map (0)
7:1	RO	1111111b	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RO	0	VC ID
30:27	RO	0	Reserved
31	RO	1	VC Enable

### C.7.1.8 Reserved Register (OFFSET 18h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.9 VC RESOURCE STATUS REGISTER 0 (OFFSET 1Ah) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### C.7.1.10 VC RESOURCE CAPABILITY REGISTER 1 (OFFSET 1Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)



### C.7.1.11 VC RESOURCE CONTROL REGISTER 1 (OFFSET 20h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	1	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.12 Reserved Register (OFFSET 24h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.13 VC RESOURCE STATUS REGISTER 1 (OFFSET 26h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### C.7.1.14 VC RESOURCE CAPABILITY REGISTER 2 (OFFSET 28h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.15 VC RESOURCE CONTROL REGISTER 2 (OFFSET 2Ch) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	2	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.16 Reserved Register (Offset 30h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.17 VC RESOURCE STATUS REGISTER 2 (OFFSET 32h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### C.7.1.18 VC RESOURCE CAPABILITY REGISTER 3 (OFFSET 34h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.19 VC RESOURCE CONTROL REGISTER 3 (OFFSET 38h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	3	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.20 Reserved Register (OFFSET 3Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.21 VC RESOURCE STATUS REGISTER 3 (OFFSET 3Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### C.7.1.22 VC RESOURCE CAPABILITY REGISTER 4 (OFFSET 40h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.23 VC RESOURCE CONTROL REGISTER 4 (OFFSET 44h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	4	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.24 Reserved Register (OFFSET 48h) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

**C.7.1.25 VC RESOURCE STATUS REGISTER 4 (OFFSET 4Ah) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**C.7.1.26 VC RESOURCE CAPABILITY REGISTER 5 (OFFSET 4Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

**C.7.1.27 VC RESOURCE CONTROL REGISTER 5 (OFFSET 50h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	5	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

**C.7.1.28 Reserved Register (OFFSET 54h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
15:0	RO	0	Reserved

**C.7.1.29 VC RESOURCE STATUS REGISTER 5 (OFFSET 56h) - WORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

**C.7.1.30 VC RESOURCE CAPABILITY REGISTER 6 (OFFSET 58h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.31 VC RESOURCE CONTROL REGISTER 6 (OFFSET 5Ch) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	6	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.32 Reserved Register (OFFSET 60h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.33 VC RESOURCE STATUS REGISTER 6 (OFFSET 62h) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved



### C.7.1.34 VC RESOURCE CAPABILITY REGISTER 7 (OFFSET 64h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RO	0	Port Arbitration Capability (Reserved for Endpoints)
13:8	RO	0	Reserved
14	RO	0	Undefined
15	RO	0	Reject Snoop Transactions (Reserved For Endpoints)
22:16	RO	0	Maximum Time Slots (Reserved For Endpoints)
23	RO	0	Reserved
31:24	RO	0	Port Arbitration Table Offset (Reserved For Endpoints)

### C.7.1.35 VC RESOURCE CONTROL REGISTER 7 (OFFSET 68h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	TC/VC Map (0)
7:1	RO	0	TC/VC Map (7-1)
15:8	RO	0	Reserved
16	RO	0	Load Port Arbitration Table (always returns 0) (Reserved For Endpoints)
19:17	RO	0	Port Arbitration Select (Reserved For Endpoints)
23:20	RO	0	Reserved
26:24	RW	7	VC ID
30:27	RO	0	Reserved
31	RW	0	VC Enable

### C.7.1.36 Reserved Register (OFFSET 6Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0	Reserved

### C.7.1.37 VC RESOURCE STATUS REGISTER 7 (OFFSET 6Eh) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Port Arbitration Table Status (Reserved For Endpoints)
1	RO	0	VC Negotiation Pending
15:2	RO	0	Reserved

### C.7.1.38 VC ARBITRATION TABLE (OFFSET 70h) – 4 DWORDs

BIT	ACCESS	DEFAULT	DESCRIPTION
2:0	RW	0	VC ID Phase 0
3	RW	0	Reserved

...

126:124	RW	0	VC ID Phase 31
127	RW	0	Reserved

## C.8. PCI Express Power Budgeting Capability

### C.8.1. Standard Device PCI Express Power Budgeting Capability

#### C.8.1.1 PCI EXPRESS POWER BUDGETING ENHANCED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0004h	Power Budgeting Extended Capability ID
19:16	RO	1h	Capability Version
31:20	RO	???h	Next Capability Offset (Points to the next capability implemented)

#### C.8.1.2 DATA SELECT REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
7:0	RW	00h	Data Select
31:8	RO	0	Reserved

**C.8.1.3 DATA REGISTER (OFFSET 08h) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
7:0	RO	00h*	Base Power (*value depends on Data Select value)
9:8	RO	00b	Data Scale
12:10	RO	000b	PM Sub State
14:13	RO	00b	PM State
17:15	RO	000b*	Type (*value depends on Data Select value)
20:18	RO	000b	Power Rail
31:21	RO	0	Reserved

**C.8.1.4 POWER BUDGET CAPABILITY REGISTER (OFFSET 0Ch) - DWORD**

<b>BIT</b>	<b>ACCESS</b>	<b>DEFAULT</b>	<b>DESCRIPTION</b>
0	RO	0	System Allocated
31:1	RO	0	Reserved

The default power budgeting capability contains the following Data Select Index and Data pairs.

Data Select Index: 0  
 Base Power 25  
 Data Scale 00b (1.0)  
 PM Sub State 000b (Default)  
 PM State 00b (D0)  
 Type 111b (Max)  
 Power Rail 000b (12 Volts)

Data Select Index: 1  
 Base Power 20  
 Data Scale 00b (1.0)  
 PM Sub State 000b (Default)  
 PM State 00b (D0)  
 Type 011b (Sustained)  
 Power Rail 000b (12 Volts)

## C.9. PCI Express Secondary PCI Express Extended Capability

### C.9.1. Standard Upstream Port PCI Express Secondary PCI Express Extended Capability (without Crosslink Support)

#### C.9.1.1 PCI EXPRESS SECONDARY PCI EXPRESS EXTENDED CAPABILITY HEADER (OFFSET 00h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0019h	Secondary PCI Express Extended Capability ID
19:16	RO	1h	Capability Version
31:20	RO	000h	Next Capability Offset (Last capability in list)

#### C.9.1.2 LINK CONTROL 3 REGISTER (OFFSET 04h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RO	0	Perform Equalization
1	RO	0	Link Equalization Request Interrupt Enable
31:2	RO	0	Reserved

#### C.9.1.3 LANE ERROR STATUS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1CS	0	Lane Error Status (x1 is the width of the test card)
31:1	RO	0	Reserved

### C.9.1.4 EQUALIZATION CONTROL REGISTER (OFFSET 0Ch) - WORD

BITS	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	0000b	Downstream Port Transmitter Preset (reserved for Upstream ports)
6:4	RO	000b	Downstream Port Receiver Preset Hint (reserved for Upstream ports)
7	RO	0	Reserved
11:8	RO	1111b	Upstream Port Transmitter Preset
14:12	RO	111b	Upstream Port Receiver Preset Hint
15	RO	0	Reserved

## C.9.2. Standard Downstream Port PCI Express Secondary PCI Express Extended Capability

### C.9.2.1 PCI EXPRESS SECONDARY PCI EXPRESS EXTENDED CAPABILITY HEADER (OFFSET 00h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
15:0	RO	0019h	Secondary PCI Express Extended Capability ID
19:16	RO	1h	Capability Version
31:20	RO	000h	Next Capability Offset (Last capability in list)

### C.9.2.2 LINK CONTROL 3 REGISTER (OFFSET 04h) - DWORD

BITS	ACCESS	DEFAULT	DESCRIPTION
0	RW	0	Perform Equalization
1	RW	0	Link Equalization Request Interrupt Enable
31:2	RO	0	Reserved

### C.9.2.3 LANE ERROR STATUS REGISTER (OFFSET 08h) - DWORD

BIT	ACCESS	DEFAULT	DESCRIPTION
0	RW1CS	0	Lane Error Status (x1 is the width of the test card)
31:1	RO	0	Reserved

### C.9.2.4 EQUALIZATION CONTROL REGISTER (OFFSET 0Ch) - WORD

BIT	ACCESS	DEFAULT	DESCRIPTION
3:0	RO	1111b	Downstream Port Transmitter Preset
6:4	RO	111b	Downstream Port Receiver Preset Hint
7	RO	0	Reserved
11:8	RO	1111b	Upstream Port Transmitter Preset
14:12	RO	111b	Upstream Port Receiver Preset Hint
15	RO	0	Reserved

# Acknowledgements

## Appendix D.

### □ AMD

- Gord Caruk
- Dean Gonzales
- Betty Luk
- Steve Manning
- Anthony Tam
- Wayne Yun

### □ Agilent

- Rick Eads
- Gordon Getty
- Thorsten Goetzelmann
- Rob Vezina

### □ IBM

- Will Atherton
- Dustin Patterson

### □ IDT

- Mike Chessin

### □ Intel

- Dan Froelich
- Manisha Nilange
- Akshay Pethe
- Marc Wells

### □ LeCroy

- Linden Hsu
- David Li
- Joseph Schachner

### □ LSI

- Richard Solomon

### □ NVIDIA

- Raymond Barrios
- Steve Glaser
- Bill Simms
- Mark Taylor

### □ PLX

- Nagamanivel Balasubramaniyan
- Satish Venugopal

### □ QuestTech

- Jerry Kinsley
- Rich Minter
- Dan Neal

### □ Tektronix

- Sarah Boen
- Jit Lim
- Steve Reinhold
- Kalev Sepp